Northumbria Research Link

Citation: Zhang, Jingtian (2019) Learning discriminative features for human motion understanding. Doctoral thesis, Northumbria University.

This version was downloaded from Northumbria Research Link: http://nrl.northumbria.ac.uk/id/eprint/42562/

Northumbria University has developed Northumbria Research Link (NRL) to enable users to access the University's research output. Copyright © and moral rights for items on NRL are retained by the individual author(s) and/or other copyright owners. Single copies of full items can be reproduced, displayed or performed, and given to third parties in any format or medium for personal research or study, educational, or not-for-profit purposes without prior permission or charge, provided the authors, title and full bibliographic details are given, as well as a hyperlink and/or URL to the original metadata page. The content must not be changed in any way. Full items must not be sold commercially in any format or medium without formal permission of the copyright holder. The full policy is available online: http://nrl.northumbria.ac.uk/policies.html

Northumbria University NEWCASTLE

University Library

Learning Discriminative Features for Human Motion Understanding

Jingtian Zhang

Department of Computer and Information Sciences
Northumbria University

This dissertation is submitted for the degree of
Doctor of Philosophy

November 2019

I would like to dedicate this thesis to my loving parents and wife.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

Word Count: 36176

Jingtian Zhang
November 2019

My Publications

- **Jingtian Zhang**, Lining Zhang, Hubert P. H. Shum, Ling Shao, Arbitrary view action recognition via transfer dictionary learning on synthetic training data, IEEE International Conference on Robotics and Automation, pp. 1678-1684, Stockholm, Sweden, May 2016.

- **Jingtian Zhang**, Hubert P. H. Shum, Jungong Han, Ling Shao, Action Recognition From Arbitrary Views Using Transferable Dictionary Learning, IEEE Transactions on Image Processing 27 (10), 4709-4723.

- **Jingtian Zhang**, Hubert P. H. Shum, Kevin McCay, Edmond S. L. Ho, Prior-less 3D Human Shape Reconstruction with an Earth Mover's Distance Informed CNN, ACM Motion, Interaction and Games, pp. 44, Newcastle upon Tyne, United Kingdom, Oct 2019.

- Worasak Rueangsirarak, **Jingtian Zhang**, Nauman Aslam, Edmond S. L. Ho, Hubert P. H. Shum, Automatic Musculoskeletal and Neurological Disorder Diagnosis With Relative Joint Displacement From Human Gait, IEEE Transactions on Neural Systems and Rehabilitation Engineering 26 (12), 2387-2396.

- Yijun Shen, **Jingtian Zhang**, Longzhi Yang, Hubert P. H. Shum, Depth Sensor based Facial and Body Animation Control, in Handbook of Human Motion, Springer International Publishing, 2016.

Acknowledgements

Firstly, I would like to express my sincerest thanks to my supervisor, Dr. Hubert P. H. Shum for his kind and delicate supervision in both research and life. He gives me constant encouragement and leads me into the field of human motion understanding. Besides, my thanks also go to Prof. Ling Shao, Dr. Jungong Han and Dr. Edmond E. L. Ho in my supervision team. They always convey novel ideas and inspire me a lot.

I would also thank my family, especially my mother, Minzhi Cai, who has provided every possible material and spiritual support for my study. Another significant person who I would like to thank to is my wife, Tiantian Chen. My achievement cannot be apart from their continued love and encouragement.

Besides, I would also like to appreciate the help and inspirations from my dear colleges, Lining Zhang, Worasak Rueangsirarak, Shoujiang Xu, Pengpeng Hu, Ying Huang as senior members, have provided many helpful supports. Also, to Yijun Shen, Shanfeng Hu, Daniel Organisciakm, Dimitris Sakkos, Kaveen Perera, Kevin Mccay, Naoki Nozawa and all of whom that could not be fully listed here, I will extend my sincere thanks to all of you.

At last, I would also like to give my thank other researchers in our laboratory: Jie Li, Yao Tan, Zheming Zuo, Baqar Abbas, I feel happy to have refreshing conversations with them on both study and life, and also learn a lot from them.

Abstract

Human motion understanding has attracted considerable interest in recent research for its applications to video surveillance, content-based search and healthcare. With different capturing methods, human motion can be recorded in various forms (e.g. skeletal data, video, image, etc.). Compared to the 2D video and image, skeletal data recorded by motion capture device contains full 3D movement information. To begin with, we first look into a gait motion analysis problem based on 3D skeletal data. We propose an automatic framework for identifying musculoskeletal and neurological disorders among older people based on 3D skeletal motion data. In this framework, a feature selection strategy and two new gait features are proposed to choose an optimal feature set from the input features to optimise classification accuracy.

Due to self-occlusion caused by single shooting angle, 2D video and image are not able to record full 3D geometric information. Therefore, viewpoint variation dramatically affects the performance on lots of 2D based applications (e.g. arbitrary view action recognition and image-based 3D human shape reconstruction). Leveraging view-invariance from the 3D model is a popular idea to improve the performance on 2D computer vision problems. Therefore, in the second contribution, we adopt 3D models built with computer graphics technology to assist in solving the problem of arbitrary view action recognition. As a solution, a new transfer dictionary learning framework that utilises computer graphics technologies to synthesise realistic 2D and 3D training videos is proposed, which can project a real-world 2D video into a view-invariant sparse representation.

In the third contribution, 3D models are utilised to build an end-to-end 3D human shape reconstruction system, which can recover the 3D human shape from a single image without any prior parametric model. In contrast to most existing methods that calculate 3D joint locations, the method proposed in this thesis can produce a richer and more useful point cloud based representation. Synthesised high-quality 2D images and dense 3D point clouds are used to train a CNN-based encoder and 3D regression module.

It can be concluded that the methods introduced in this thesis try to explore human motion understanding from 3D to 2D. We investigate how to compensate for the lack of full geometric information in 2D based applications with view-invariance learnt from 3D models.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

Understanding human motion is based on studying global human motion patterns, rather than on analysing local patterns such as facial expressions and hand gestures. Human motion understanding based on multiple kinds of inputs is a field of research of increasing importance. A lot of research has drawn much inspiration from human motion, ranging from action recognition to human animation rendering. Together with tools from biomechanics, human motion understanding enables our efforts to explore natural human motion, leading to improvements in treatments for patients with neurological and musculoskeletal disorders, and facilitating the development of human-inspired robots. Using computer vision and computer graphics technologies, we aim to gain fundamental insight into natural human motion and understand the mechanisms that lead to improved performance in action recognition and 3D human shape reconstruction.

With the development of motion capture technology, more accurate 3D human motion information (e.g. skeletal data) can be acquired to help us to understand human motion. Skeletal data acquisition generates a huge amount of high-dimensional data. Machine learning has appeared as a good way to extract information to describe the properties of a collection of high-dimensional motion-captured data. It can avoid a lot of manpower and mistake from doctor's diagnosis by automatically diagnosing motion disorder with machine learning technologies.

Vision is an essential perceptive approach for understanding human motion in images and videos. The ultimate goal of computer vision is to understand the scene and object correctly through various steps of acquiring, processing, analysing and interpreting different kinds of information obtained by different types of sensors. We focus on the computer vision technologies used in human motion understanding. Thanks to the convenience of the shooting devices, 2D videos and images are the most popular media to record human motion. However, 2D videos and images are not able to capture full 3D geometric information because of the

limitation of a single shooting angle. 2D appearance dramatically changes with a viewpoint changing. Therefore, in this thesis, we leverage view-invariance learnt from 3D models to solve arbitrary view action recognition and image-based 3D human shape reconstruction problems.

Creating suitable and abundant training data for supervised learning is a very challenging problem in computer vision. How do we efficiently generate such training data? The dominant data acquisition method in visual recognition is based on web data and manual annotation. However, for many computer vision problems related to 3D models, this approach is not feasible because the number of high-quality 3D models that meet the requirements on the web is far less than the number of images. In this thesis, we use computer graphics technologies to generate synthetic human motion data for training based applications. We also demonstrate the benefit of learning schedules that use different types of data at selected stages of the training process.

1.1 Motivations

The research goal of this thesis is to leverage view-invariance learnt from 3D human body models to solve some 2D human motion understanding applications. We develop learning methods to advance automatic analysis and interpretation of human motion from different perspectives, based on both 2D and 3D information, such as images, video and mocap data. For this purpose, we propose machine learning models to learn human motion features, and show their efficiency on a set of fundamental tasks.

Explicitly, we first apply machine learning technology to 3D skeletal data to learn features for identification of gait disorders. Then, we utilise the knowledge and experience from 3D data to compensate for the lack of full geometric information in 2D based applications (e.g. arbitrary view action recognition and image-based 3D human shape reconstruction).

Gait disorder analysis Human motion understanding can assist doctors with diagnosing abnormal motion. For example, gait analysis is a popular method for diagnosing musculoskeletal and neurological disorders, but determining abnormal gaits can be challenging. Gait disorders are among the most common causes of balance problems in older people and often lead to injury, disability, loss of independence, which result in poor quality of life. Manually analysing gait data is a labour intensive process, and its accuracy depends on the expertise of the doctors performing the analysis. This is because there are no clearly accepted standards to evaluate the gait of older people. Previous work investigates detailed biomechanical analysis to study musculoskeletal and neurological disorders. Regardless of the methods used, manually analysing gait data is a labour intensive process, and its accuracy

depends on the experience of the doctors. Combined with machine learning and feature selection methods, we propose an automatic framework for identifying musculoskeletal and neurological disorders among older people. This can classify the motion into four classes: healthy, muscle weaknesses, joint problems or neurological defects.

Arbitrary view action recognition 2D video based human action recognition plays an important role in human motion understanding, it has attracted a lot of attention in area such as security surveillance and human-computer interaction. However, most of these approaches are only effective for single view action recognition, and the recognition performance degrades significantly when the viewpoint is changed. This is because the appearance of a particular action from different viewpoints varies dramatically. Leveraging the view-invariance of 3D models is a general approach to solve this problem, which may give results that are highly dependent on the quality of the 3D model. Therefore, combined with computer graphics technology, we generate high-quality 3D models and propose a smooth view-invariance transfer system between 3D models and 2D images.

3D human shape reconstruction 3D human shape reconstruction from a single image is a highly under-determined problem, requiring strong prior knowledge of plausible 3D human shapes. Previous work uses a prior parametric model and generate the reconstructed 3D model based on specific parameters (e.g. joint locations, pose categories). However, due to the complexity of the human body shape and the variety of the different body sizes, relying on the parametric model can not precisely recover details of body shape. In contrast to most current methods that compute 3D joint locations, we utilise computer graphics knowledge to produce a richer and more useful point cloud based representation, which can retain as many details as possible.

1.2 Problems Definition and Methodology Overview

This section introduce three problems in human motion understanding that we address in this thesis. We first define the problems, and then introduce the corresponding methodology overview.

1.2.1 Gait disorder analysis

Musculoskeletal and neurological disorders are common devastating companions of ageing, leading to a reduction in quality of life and increased mortality. Gait analysis is a popular method for diagnosing these disorders. However, manually analysing the motion data is a labour-intensive task, and the quality of the results depends on the experience of the doctors.

Methodology Overview We propose an automatic framework for classifying musculoskeletal and neurological disorders among older people based on 3D motion data. We also propose two new features to capture the relationship between joints across frames, known as 3D Relative Joint Displacement (3DRJDP) and 6D Symmetric Relative Joint Displacement (6DSymRJDP), such that relative movement between joints can be analyzed. To optimize the classification performance, we adapt feature selection methods to choose an optimal feature set from the raw feature input. Experimental results show that we achieve a classification accuracy of 84.29% using the proposed relative joint features, outperforming existing features that focus on the movement of individual joints. Considering the limited open motion database for gait analysis focusing on such disorders, we construct a comprehensive, openly accessible 3D full-body motion database from 45 subjects.

1.2.2 Arbitrary View Action Recognition

Most existing work leverages view-invariance provided by 3D models to realize cross-view or arbitrary-view action recognition. Traditionally, simplified cylindrical models are used [1, 2], which do not generate realistic movement appearance. High-quality reconstruction models are proposed by calculating them from multi-view 2D videos [4]. In order to increase the system robustness to viewpoint changes, training data is forced to cover as much 2D data projected along as many viewpoints as possible. All these approaches suffer from the following problems: (1) The recognition accuracy is highly related to the quality of 3D models. Some human appearance information could be lost due to the unrealistic 3D reconstruction and the simplified cylindrical model; (2) Despite the effort to project the 3D model into as many viewpoints as possible, these discrete projection angles will inevitably result in the loss of 3D geometric information. A large number of 2D projections also requires larger system capacity and training cost.

Methodology Overview We propose a novel end-to-end framework to jointly learn a view-invariance transfer dictionary and a view-invariant classifier. The result of the process is a dictionary that can project real-world 2D video into a view-invariant sparse representation, as well as a classifier to recognize actions with an arbitrary view. The main feature of our algorithm is the use of synthetic data to extract view-invariance between 3D and 2D videos during the pre-training phase. This guarantees the availability of training data, and removes the issue of obtaining real-world videos in specific viewing angles. Additionally, for better describing the actions in 3D videos, we introduce a new feature set, called 3D dense trajectories, to effectively encode extracted trajectory information on 3D videos. Experimental results on the IXMAS, N-UCLA, i3DPost and UWA3DII datasets show improvements over existing algorithms.

1.2.3 Image-based 3D Human Shape Reconstruction

3D human shape reconstruction from 2D RGB images is challenging and ambiguous. All methods try to address the ambiguity in different ways. Most methods adopt prior knowledge to control the ambiguity. For example, prior shape model can enforce anthropometric constraints on bone lengths, and prior pose information can keep 'possible' poses and eliminate 'impossible' ones. However, since the geometry of the human body shape is very complex, and body size is different across people, using only a parametric model can not precisely recover details of body shape.

In contrast to most existing studies that depend on 2D and 3D locations, we produce a richer and more useful point cloud based representation, which is able to retain as much detail as possible. The main objective is to minimize the loss between the ground truth and the predicted 3D human shape model.

Methodology Overview We provide a solution that is fully automatic, which estimates a 3D point cloud capturing human shape from a 2D image without any prior parametric model. An image S is passed through a convolutional encoder. This is sent to a 3D regression module that infers the latent 3D representation of the human that minimizes the Earth Mover's Distance (EMD) between the ground truth and the predicted 3D human body shape.

1.3 Thesis Structure

The structure of the rest of the thesis is as follows: I first review related work in the second chapter Chapter 2. This part covers the state-of-art techniques from various fields that are related to the topic. Then in the following three chapters, Chapter 3, Chapter 4 and Chapter 5, I detail our solutions of the three problems raised above. Finally, I conclude the thesis and discuss some future work in Chapter 6.

1.4 Summary

2D videos and images are the most popular media to record human motion. However, 2D videos and images are not able to capture full 3D geometric information because of the limitation of a single shooting angle. 2D appearance dramatically changes with the viewpoint changing. Therefore, in this thesis, we leverage view-invariance learnt from 3D models to solve 2D based computer vision problems. First, we propose a framework to automatically identify the type of gait disorder directly on 3D skeletal data. Second, a transfer dictionary learning system is built to transfer the view-invariance from 3D models to 2D

videos smoothly. Finally, a deep shape recovery network is designed for 3D human shape reconstruction from a 2D image with 3D point cloud based models.

Key words: Human motion understanding, Discriminative feature learning, Gait disorder diagnosis, Arbitrary view action recognition, 3D human shape reconstruction, Machine learning, Computer vision, Computer graphics, Motion analysis.

Chapter 2

Literature Review

In this chapter, we introduce some background knowledge of the three applications tried to solve in the thesis. We first review frequently-used techniques for gait analysis, such as gait feature selection methods and automatic diagnosis methods. Furthermore, basic techniques for arbitrary view action recognition are introduced (e.g. 3D examplar based methods and interest point based methods). Finally, we discuss some techniques used in 3D human shape reconstruction.

2.1 Motion Analysis for gait disorder diagnosis

Extensive research efforts have been made towards gait disorder analysis. In this section, we first discuss the different features adopted in gait analysis. We present a summary of machine learning methods for improving gait analysis, including feature selection algorithms and gait pattern classification algorithms.

Gait disorders are among the most common causes of balance problems in older people [5] and often lead to injury, disability, loss of independence, which result in poor quality of life. The prevalence of these conditions is expected to rise dramatically as the population ages. At least 30% of people aged 65 and older report difficulty walking three city blocks or climbing one flight of stairs, and approximately 20% require the use of a mobility aid to move [6]. Musculoskeletal and neurological disorders are some of the major reasons for an abnormal gait. They have been detected in approximately 25% of people between 70 and 74 years of age, and in nearly 60% of those between 80 and 84 years of age [7].

Gait analysis is a popular method for diagnosing the musculoskeletal and neurological disorders, but determining abnormal gaits can be challenging. This is because there are no clearly accepted standards to evaluate the gait of older people [8]. While some research investigates time-distance variables (e.g. walking speed, step length) [9], others carry

out detailed biomechanical analysis (e.g. joint rotation, joint position) [10] to investigate musculoskeletal and neurological disorders. Regardless of the methods used, manually analysing gait data is a labour intensive process, and its accuracy depends on the experience of the doctors performing the analysis.

Automatic diagnosis of musculoskeletal and neurological disorders using machine learning technology has shown to be effective in reducing the manpower required for gait analysis and ensuring the reliability of the diagnosis results [11]. Holzreiter and Kohle apply artificial neural networks (ANNs) for the classification of normal and pathological gaits using foot-ground reaction forces [12]. Barton and Lees apply ANNs to differentiate different types of gaits using joint angles [13] to distinguish different gait patterns. Begg et al. extract the Minimum Foot Clearance (MFC), the shortest distance from the floor to the toes during the swing phase, and use it to input to a support vector machine (SVM) classifier [14] for gait balance classification. Khandoker et al. combine multiple features for gait analysis [15]. They concatenate a selection of extracted statistical values from the MFC histogram (e.g. Mean, Standard deviation, Maximum and Minimum) as identification features for the balance impairment classification for older people. Despite many successes, the majority of existing work only evaluates features extracted from individual joints independently, and ignores the relationship between different joints that could be useful in gait analysis. Also, selecting a subset of joints based on expert knowledge [14, 15], may not be optimal for gait classification from the data point-of-view. Concatenating a large number of features into a single feature vector could have an adverse effect on classification accuracy, due to noise in some features, and could cause low system efficiency.

2.1.1 Features for Gait Analysis

Gait features are important for objective gait assessment and analysis. The core of many contemporary features for gait analysis is the measurement of joint kinematics and kinetics, such as the Conventional Gait Model and the Cleveland Clinic Model [16]. Among many gait features, symmetry is an important gait characteristic and is defined as a perfect agreement between the actions of the two lower limbs [17]. To calculate symmetry, mobility parameters (e.g. single joint rotation) and spatiotemporal parameters (e.g. step length) can be used [18]. Since it is difficult to diagnose the class of disorder solely based on asymmetric gait, balance and walking stability are also used. Multiple balance and stability measures include RMS acceleration, jerk (the derivative of acceleration), sway (a measure on how much a person leans his/her body), step and stride regularity and variability [19]. Mobility parameters such as cadence and step length are also important indicators to quantify gait [20]. These kinds of balance and walking stability parameters are hand-crafted and require expert knowledge. In

our research, we propose a new, generic feature based on relative joint information, which is an important addition to the currently developed methods for identifying gait abnormalities.

2.1.2 Gait Feature Selection Methods

Simply concatenating gait features typically results in high-dimensional data, and some dimensions may not be relevant for the problem in hand. It is therefore advantageous to identify the important features for gait analysis, thereby removing features that convey little or redundant discriminatory information. Some techniques [21, 22] involve choosing a subset of original features that can represent the original data under certain criteria. They mainly use conventional dimensionality reduction or statistical tools, such as principal component analysis (PCA) [23, 24] and F-score [25]. Robnik-Šikonja and Kononenko propose ReliefF, which weights different features by maximizing the distance between the data belonging to different classes [26]. Yang et al. propose Neighborhood Component Analysis (NCA) to learn a feature weighting vector by maximizing the expected leave-one-out classification accuracy using a regularization term [27]. In our research, we adapt F-score, ReliefF and NCA for feature selection, and evaluate their respective performance.

2.1.3 Automatic Diagnosis Methods

Many machine learning algorithms have been adapted to automatically diagnose gait disorder. In [28], non-hierarchical cluster analysis is used to categorize four subgroups based on the temporal-spatial and kinematic parameters of walking. Similarly, hierarchical cluster analysis is used in [29], identifying three groups of subjects with homogeneous levels of dysfunction. Artificial neural networks (ANN) are used in [30] to classify a post-stroke subject's gait into three categories based on the types of foot positions on the ground at first contact. In [31, 32], the ability of ANN and Support Vector Machine (SVM) to distinguish gait patterns for Parkinson Disease is discussed. Gait features from wavelet analysis and kinematic parameters are extracted, which are passed to the SVM for classification [15]. We also utilize SVM in our method, as it has great usability in clinical routines without introducing complex apparatus.

2.2 Arbitrary view action recognition

Understanding human motion in 2D videos has attracted a lot of attention in security surveillance and human-computer interaction. Various spatio-temporal appearances generated from the movements can be considered as the feature descriptors for action recognition. These

include spatio-temporal pattern template [33], spatio-temporal interest points [34–37], shape matching [38, 39] and motion trajectories based descriptors [40–43]. Among them, dense trajectories based methods have achieved state-of-the-art results by extracting densely sampled trajectories-aligned descriptors in the optical flow fields. Deep learning networks have also achieved significant success in the 2D action recognition area [44–47]. These methods can automatically learn spatial-temporal feature representations and identify different action categories. However, [44–47] are only effective for single view action recognition and the recognition performance degrades significantly when the viewpoint is changed. This is because the appearances of a particular action from different viewpoints vary dramatically, which results in dissimilar trajectories.

As a result, cross-view action recognition is proposed for bridging the appearance differences between different viewpoints. The main idea is to transfer the knowledge from the source view to the target view, allowing the system to recognize actions from a view that is not included in the training set. Li et al. presented a dynamics-based feature called hankelet that can capture the invariant property in viewpoint change using short tracklets for cross-view recognition [33]. Wang et al. used an AND-OR graph representation to compactly express the appearance and motion variance during viewpoint changes [48]. Zhang et al. constructed a continuous path between the target view and the source view to facilitate cross-view action recognition [38, 39]. Farhadi et al. generated the same split-based features for correspondence video frames from both training and testing views [49]. Such systems are computationally expensive as they not only require feature-to-feature correspondence, but also require mapping between the split-based and the original feature. Liu et al. used a bipartite graph to model the relationship between the two codebooks from the source view and target view [50]. Wang et al. proposed a Statistical Translation Framework (STF) to estimate the transfer probabilities of the visual words from the source to target views [51]. Huang et al. built a correlation subspace to produce joint representation from different views by using canonical correlation analysis [52]. In spite of discovering the correspondence between codebooks from two or more different views, the above approaches cannot guarantee that videos captured from different views share similar features. Also, all these methods require viewpoint information for both source view and target view, which is usually not available in practical applications.

As a solution, arbitrary-view action recognition is proposed, in which viewpoint information is not required during testing and action from unseen views can be recognized. The main idea is to remove view-dependent information from the feature representation. Previous attempts to realize arbitrary-view action recognition have met with varying levels of success. Lv et al. use a graphical model to calibrate 2D key poses of actors to represent

3D surface models for arbitrary view action recognition. However, the motion information for recognizing actions may not be well captured [39]. Weinland et al. propose to recognize human actions by estimating 3D exemplars from a single 2D view angle using the hidden Markov model [53]. However, reconstructing these 3D exemplars from a single view is unreliable. Also, detailed action information may be lost as only discrete samples of silhouette information are used. Yan et al. present a 4D (i.e., 3D spatial and 1D temporal dimensions) action feature using the time-ordered 3D reconstruction of the actors from multi-view video data [4]. The recognition accuracy depends heavily on the performance of the 3D reconstruction, and the framework requires training data to be captured from carefully designed viewpoints. Gupta et al. propose to project the 3D motion capture sequence in the 2D space and explore the best match of each training video using non-linear circular temporary encoding [1]. However, since discrete 2D projection, instead of full 3D information, is used for training, the accuracy depends on the number of projected views. Rahmani et al. propose R-NKTM to transfer knowledge of human actions from any unknown view to a shared high-level virtual view by finding a non-linear virtual path that connects the views [54]. They generate the training data by projecting the 3D exemplar to 108 virtual views. The use of so many projected views results in enhanced system performance, but result in a computationally expensive training process. Ideally, we would like to have a framework that relies on easy-to-obtain training data and performs robustly in runtime.

The general process for view-invariant action recognition can be divided into three major parts: (1) Synthesized 3D exemplars are used for producing the 2D videos covering as many viewpoints as possible. (2) Then, the feature extraction methods, especially some interest points and trajectory-based feature extraction methods, describe the action on the 2D videos. (3) Finally, transfer learning algorithms are used to transfer the action information across different views in order to realize view-invariant action recognition. Here, previous work related to these three major processes is introduced.

2.2.1 3D Exemplar-based Methods

One popular idea is to utilize 3D exemplars for view-invariant feature extraction and description. Some researchers use the static 3D exemplars. For example, Ankerst et al. propose the histogram of shape [55] which is very similar to the 3D shape context proposed by Korgen et al. [56]. Subsequently, Huang et al. combine the histogram of shape with color information [57]. All these methods are mainly based on static descriptors such as poses and shape, while the state-of-the-art descriptors integrate static descriptors with motion information.

Instead of relying only on static feature, some researchers utilize the derivative of static descriptors over time in order to capture the temporal information by simply accumulating

static descriptors, applying sliding windows, or tracking human pose information [58–60, 53]. Cohen et al. [61] present a 3D human shape model for view-invariant human identification. Later, this 3D human shape model was developed by Pierobon et al. [60] for human action recognition. Weinland et al. propose the Motion History Volume (MVH) as a 3D extension of Motion Histogram Images (MHIs) [59]. MHV is calculated by accumulating human postures over time in cylindrical coordinates. A different strategy is proposed by Yan et al. [4], where they develop a 4D action feature model (4D-AFM) for arbitrary view action recognition based on spatio-temporal volumes (STVs). However, the performance of the above 3D exemplars-based systems is strictly limited to the result of 3D reconstruction. Normally, the reconstructed 3D exemplars are not very realistic.

Other researchers construct the 3D exemplar with the aid of depth sensors. Zhang et al. present a low-cost descriptor called 3D histogram of textures (3DHoTs) to extract discriminative features from a sequence of depth maps [62]. They combine depth maps and texture description by projecting depth frames onto three orthogonal Cartesian planes to describe the salient information of a specific action. Liu et al. present a multi-scale energy-based Global Ternary Image (GTI) representation, which efficiently encodes both the spatial and temporal information of 3D actions [63]. Skeleton information can easily be collected from the depth map. Liu et al. propose a sequence-based transform method, which maps skeleton joints into a view-invariant high dimensional space [64]. They use color images to visualize this space, and apply CNN to extract deep features from these enhanced color images. Wang et al. realize non-rigid reconstruction and motion tracking without any template using a single RGB-D camera [65]. Jia et al. present a tensor subspace, whose dimension is learned automatically by low-rank learning for RGB-D action recognition [66]. Kong et al. propose a discriminative relational feature learning method for fusing heterogeneous RGB with depth modalities and classifying the actions in RGB-D sequences [67]. Even though the depth information has a superior descriptive ability on 3D exemplars, most videos in the real world are captured without depth information. Therefore, we focus on techniques for extracting 3D information from RGB only videos, which has more potential applications.

2.2.2 Interest Points and Trajectory-based Methods

To better describe the spatio-temporal interest points, Dollar et al. present descriptors on brightness, optical flow and gradient information [34]. The SIFT descriptor is extended to the spatio-temporal interest points by Scovanner et al. [68]. Willems et al. extend the SURF descriptor to the video domain by computing weighted sums of response of Haar wavelets [69].

Due to the fact that spatio-temporal interest points are at fixed location in the video, only base on interest points descriptors cannot capture motion information in the video. In contrast, trajectories track the given interest point over time, so it can capture the motion information. Messing et al. extract trajectories by tracking Harris3D interest points with a KLT tracker [70]. They use a sequence of log-polar quantized velocities to represent trajectories. Matikainen et al. extract trajectories with a standard KLT tracker, then they cluster these trajectories for the action classfication [71]. Sun et al. match SIFT descriptor between two frames to compute trajectories [72]. Later, they combine both SIFT matching and KLT tracker to extract long-duration trajectories [73]. Wang et al. compute trajectories by tracking the interest points in the optical flow field, then they compute Histogram of Gradient (HOG), Histogram of Optical Flow (HOF) and Motion Boundary Histogram (MBH) to model the action in the video [41]. However, the optical flow field is just a 2D approximation of the 3D motion field, and cannot accurately describe the 3D motion information.

2.2.3 Transfer Learning and Dictionary Learning

Transfer learning has been widely used in cross-domain action recognition problems to take knowledge gained from one dataset and apply it to a different but related one. Liu et al. present a simple-to-complex action transfer learning model (SCA-TLM) for complex human action recognition [74]. It improves the performance of complex action recognition by leveraging the abundant labeled simple actions. In particular, it optimizes the weight parameters, enabling complex actions to be learned and to be reconstructed by simple actions. Xu et al. propose a novel dual many-to-one encoder architecture to extract generalized features by mapping raw features from source and target datasets to the same feature space [75]. Rahmani et al. propose R-NKTM to transfer knowledge of human actions from any unknown view to a shared high-level virtual view by finding a non-linear virtual path that connects the views [54].

Recently, dictionary learning for sparse representation has been successfully applied in many computer vision applications, such as image de-noising [76] and face recognition [77]. With an over-complete dictionary, input signal can be approximately represented by a sparse linear combination of items in the dictionary. Previously, many methods [78] have been presented to learn such a dictionary based on different criteria. Among them, K-Singular Value Decomposition (K-SVD) [79] is a typical dictionary learning method that uses the K-means clustering algorithm for optimizing dictionary items to learn an over-complete dictionary. Even though the K-SVD method has re-constructive ability, due to the unsupervised learning process, its discriminative ability has not been considered. Later, [80] proposed a dictionary transformation method to transform the dictionary from one domain to

another. It can handle the problem that the testing instances are different from the training instances. In addition, they use correspondences between the source view and the target view to construct pairwise dictionaries for the cross-view action recognition problem. Zheng et al. represent the videos in each view using a view-specific dictionary and the common dictionary. More importantly, it encourages the set of videos taken from different views of the same action to have the similar sparse representations [81].

Unlike the above approaches, our approach simultaneously learns pairwise dictionaries and a classifier while considering re-constructive ability, discriminative ability and domain adaptability during the dictionary learning process. The data in 3D source domain and 2D target domain are with completely different formats. View-invariance from 3D data can be smoothly transferred to 2D data with jointly optimizing the pairwise dictionaries.

2.3 Image-based 3D Human Shape Reconstruction

Our goal is the accurate estimation of a 3D human shape from a single 2D image. Whilst there is a large amount of relevant literature relating to estimating human body posture and shape from multiple camera images or video sequences, in this section, we focus on static image methods due to the more practical hardware setup. As data is essential to train a deep network, we also review methods to obtain 3D human surface data. Estimating the 3D body shape of humans is an increasingly important problem due to its application in surveillance monitoring [82], motion analysis [83], and even online clothing shopping [84]. These applications typically require analysis of the body shape based upon an observed input, which consists of information from both part size and posture.

2.3.1 Prior-based Reconstruction

Early works in the field rely on shape silhouettes to help estimating 3D shapes. Guan et al. [85] estimate the 3D posture based upon manually marked 2D joint positions. They project the SCAPE [86] model into the image and use GrabCut to segment the image [87]. They then use the SCAPE shape and posture to generate a variety of features including shading cues, image edges and a shape silhouette. Sigal et al. [88] calculate the shape feature from silhouettes. They then generate the 3D body shape and posture from those features. A generative network is fit to the image silhouettes to construct a fully automatic framework. They show some results using perfect silhouettes, however, they require both manually provided correspondence and a known segmentation.

Parametric models are used heavily in the field as priors. Zhou et al. [89] also use a parametric model of posture and body shape to clearly segment the silhouette. Similarly, Chen et al. [90] fit a prior parametric model of posture and the human body to extracted silhouettes manually.

Loper et al. [91] propose to apply PCA to a set of captured 3D human shape to learn one of the most popular parametric prior model called SMPL. As the dataset is captured from real-world 3D scans, it is relatively small and costly to annotate. Macard et al. [92] propose to combine the use of a camera and wearable inertia sensors to estimate 3D poses. They jointly optimise the pose, the heading drift and the estimated camera pose with the help of SMPL. Kanazawa et al. [93] combine the use of SMPL shape parameters, skeleton pose parameters and camera project parameters to construct an end-to-end network for 3D body mesh reconstruction. Similarly, Zhang et al. [94] use the SMPL parametric model as prior with a focus on estimating the 3D body shape of clothed human forms. Varol et al. [95] propose an end-to-end network that provides information about the hierarchy of the body, and utilize SMPL to reconstruct the volumetric body-part. Yu et al. [96] propose a method for real-time human performance capture which simultaneously reconstructs the body geometry, non-rigid motion and the inner human body shape from a single depth camera. They produce a double layer representation that is composed of an SMPL-based prior driven inner surface that represents the human body, and a prior-less point cloud for the outer cloth surface,implementing a loss function to correlate the two layers.

Apart from parametric priors, index models are also commonly used as prior. Such models are particularly popular in the field of joint detection with a set of indexed joints. Kulkarni et al. [97] estimate 2D joint locations based on a CNN and uses a monocular video sequence to find the 3D posture. Zhou et al. [98] propose a 2D pose detector and optimise the 3D posture by rejecting outliers. Yasin et al. [99] use the detected 2D joints to find the nearest 3D postures in a mocap dataset. Ionescu et al. [100] predict each 3D body part from the image before combining them. Simo-Serra et al. [101] propose a probabilistic model that matches the 3D posture to the 2D image features. The concept of indexed model is extended by Lassner et al. [102] to represent key landmarks on the human shape surface. Since the landmarks are indexed, the system can consider the human shape as an ordered list of 3D landmark positions.

As such prior constrain the representation of fine details and topological differences, we attempted to challenge this problem without a prior model.

2.3.2 Human Surface Data Acquisition

A major challenge faced in training deep networks is the availability and variety of the training data. Whilst it is easy to obtain real world 2D images in the wild, it is costly to annotate these images and to obtain 3D ground truth data.

Loper et al. [91] capture 3D human surface with minimum clothing for learning a set of shape and pose prior, known as SMPL. Extending SMPL, Lassner et al. [102] and Hesse et al. [83] propose methods to obtain 3D body model fits for multiple human pose datasets; however, these approaches still only annotate 2D joint positions. Pons-Moll et al. [103] and Starck et al. [104] also propose 3D capture methods, and whilst the techniques used generate good quality 3D meshes, they provide little variety as the cost of capturing the data is high.

To relieve the cost of database creation, Varol et al. [105] propose the use of a synthetic database for machine learning of human shape. They present the SURREAL dataset that includes both an artificial human and a real world background; however the 3D mesh mesh surface is rather simple and the integration with the background is unrealistic due to lighting and scene layout. Zhang et al. [82] propose the creation of a synthetic 2D and 3D training dataset using transfer dictionary learning. They create highly realistic human model for better transfer learning performance.

2.3.3 Point-cloud Representations

It is a challenging problem to perform machine learning on 3D shapes represented with an unordered point cloud. Qi et al. [106] proposed to tackle the problem with a multistage pipeline, in which shapes are first classified in order to apply proper segmentation for part-based shape modelling. The classification driven segmentation process can be considered as a type of prior. Qi et al. [107] extend the method to include a hierarchical neural network structure such that fine-grained patterns can be better represented.

Fan et al. [108] propose one of the few prior-less point cloud deep neural networks for 3D reconstruction from a single image input. An advantage is that the system can model a good level of detail for a large variety of objects. However, their focus is on rigid objects such as chairs and cars, and it is unclear if they can model the complex deformation of a human body during movement. We propose a different network structure that focuses exclusively on human shape reconstruction.

Chapter 3

Motion Analysis for gait disorder diagnosis

In this chapter, we initially explore the 3D human motion data and 3D feature learning system by examining a gait motion analysis problem directly based on 3D skeletal data. Research on human motion understanding is able to assist doctors in identifying abnormal motions.

Musculoskeletal and neurological disorders are common devastating companions of ageing, leading to a reduction in quality of life and increased mortality. Gait analysis is a popular method for diagnosing these disorders. However, manually analysing the motion data is a labour-intensive task, and the quality of the results depends on the experience of the doctors. In this paper, we propose an automatic framework for classifying musculoskeletal and neurological disorders among older people based on 3D motion data. We also propose two new features to capture the relationship between joints across frames, known as 3D Relative Joint Displacement (3DRJDP) and 6D Symmetric Relative Joint Displacement (6DSymRJDP), such that relative movement between joints can be analyzed. To optimize the classification performance, we adapt feature selection methods to choose an optimal feature set from the raw feature input. Experimental results show that we achieve a classification accuracy of 84.29% using the proposed relative joint features, outperforming existing features that focus on the movement of individual joints. Considering the limited open motion database for gait analysis focusing on such disorders, we construct a comprehensive, openly accessible 3D full-body motion database from 45 subjects.

3.1 Introduction

In this project, we use features based on relative movement information for gait analysis. Previous research in analysing the distance between the left and right feet is a solid support for the effectiveness of relative information [109], but is limited, since other joint pairs are not considered. Here, we propose two comprehensive features that capture the relative information between all pairs of joints across frames, which we call as the 3D Relative Joints Displacement (3DRJDP) and the 6D Symmetric Relative Joint Displacement (6DSymRJDP). Compared with existing methods that utilize features extracted from individual joints, our proposed method is capable of evaluating the relationships of joints for all joint pairs, providing a holistic view of all interactions. Such comprehensive information allows us to develop methods that outperform existing work in gait disorder classification.

We also use feature selection algorithms from the field of machine learning to further improve system performance. Concatenating all possible features for gait analysis has an adverse effect, as some features are noisy or even irrelevant. Manually selecting the features based on expert knowledge is sub-optimal. Therefore, we evaluate and adopt a number of feature selection algorithms to select an optimal feature set from the input features. In particular, during feature selection, we consider the whole temporal series of the relative features from two joints as a unit. This allows us to create human-understandable results and ensure a reasonable system training cost. We develop a fully automated framework to evaluate the quality of the features using F-score, Neighborhood Component Analysis (NCA) and ReliefF. The system then iteratively selects the best I features that maximize the classification accuracy.

Finally, in view of the limited available resources for 3D gait analysis, we construct a comprehensive 3D motion database captured from 45 older people, annotated with the subjects' anonymised medical history. Based on the agreed decision from 3 medical doctors, the subjects are diagnosed as entire healthy, having muscle weakness (e.g. muscle strain, underdeveloped muscles), having joint problems (e.g. degenerative joint disease, osteoarthritis), or having some neurological defect (e.g. Parkinson's disease, Alzheimer's diseases). All three classes of disorder result in movement difficulties, which may appear similar at the movement level. However, the underlying causes are very different. The database is freely available for academic and research use for free, and is downloadable from our research website (http://hubertshum.com/info/tnsre2018.htm).

We present four main contributions in this chapter:

- We propose an automatic framework for identifying musculoskeletal and neurological disorders among older people based on 3D motion data.

- We propose two new features called the 3D Relative Joints Displacement (3DRJDP) and the 6D Symmetric Relative Joint Displacement (6DSymRJDP) to capture the relationship of joint pairs across frames.
- We adapt feature selection methods including F-score, Neighborhood Component Analysis and ReliefF, for choosing an optimal feature set from the input features to optimize classification accuracy.
- We construct an openly accessible, comprehensive 3D gait database with the anonymised medical history of the subjects. The subjects are diagnosed as healthy, muscle weaknesses, joint problems and neurological defects by 3 medical doctors.

3.2 System Overview

An overview of the proposed system is presented in Fig. 3.1. We capture the 3D motion of walking from the subjects and collect their anonymised medical histories. We pre-process the captured data using inverse kinematics and dynamic time warping. We extract different types of features from the captured data, including 3DRJDP and 6DSymRJDP we proposed, as well as other common joint-based features. We adopt 3 feature selection methods and evaluate their effectiveness. Finally, the selected feature set is passed into a gait classifier.

Fig. 3.1 The overview of our automatic method for gait disorder diagnosis.

3.3 Data Collection

In this section, we first introduce the information about the invited subjects. Then, we give details on our process to capture motion data.

Table 3.1 The causes of disorder and subjects' statistics.

Muscle Weakness	Joint Problem	Neurological Defect	Healthy
18	4	13	10

	Min.	Max.	Average
Age	61	91	70.26
Weight	33	67	54.06
Height	138	198	154.26
MMSE-Thai*	15	30	24.16
FES-I**	16	26	18.96

* Mini-Mental State Examination ** Thai Fall Efficacy Scale-International

3.3.1 Subjects

The data was collected from a total of 45 subjects in a voluntary manner, whose protocol was agreed by the Faculty of Associated Medical Sciences Ethics Committee at Chiang Mai University. The experiment was conducted in Chiang Mai, Thailand. Applicants were Thai older people living in dwelling communities and nursing homes in Chiang Mai.

Three medical doctors from the Faculty of Associated Medical Sciences, Chiang Mai University attended the assessment. They diagnosed the subjects and agreed on the respective causes of the gait disorder, which were classified into healthy, muscle weakness, joint problem and neurological defect. This classification scheme was suggested as it was very useful as an initial diagnosis. With our system, patients can be efficiently and accurately directed to the relevant departments for further evaluations, in order to understand the specific causes of the disorder. In developing countries where there are limited budget and manpower for health-care, such an initial diagnosis scheme can effectively screen patients and relieve the stress from the front line. We discuss the possibility of employing depth camera based motion sensing system for markerless motion capturing or even everyday movement tracking in Section 3.7.4.

The voluntary applicants were screened and approved by medical experts using standard clinical tests. Applicants were only included if they satisfy the following requirement: (1) They could walk without any assistance from physiotherapist or gait aids for at least 10 meters. (2) They had no cognitive impairment as tested by the Mini-Mental State Examination (MMSE-Thai 2002), i.e., MMSE-Thai ≤ 14 for older people who were uneducated, MMSE-Thai ≤ 17 for older people who graduated in primary school level, and MMSE-Thai ≤ 22 for older people who graduated in high-school level or above [110]. (3) They had no fear

of falling as tested by Thai Fall Efficacy Scale-International (FES-I), i.e., Thai FES-I < 23 [111]. (4) They had no other medical history that affected walking than those we considered. (5) They had no pain while walking.

We performed a randomly sampled, population-based study. In particular, we randomly selected 45 older people with ages ranging between 61 and 91 from the approved list of the applicants, which resulted in 5 male subjects and 40 female subjects. The gender bias in the database reflects that of the voluntary applicants. The details information of the subjects including the causes of the disorder, age, weight, height, MMSE-Thai and FES-I are summarized in Table 3.1.

3.3.2 Data Acquisition

Direct interviews with the subjects were conducted using a structured questionnaire [112] before their walking sessions. Fig. 3.2 shows the questions in the questionnaire. The collected data covered known diseases, medication history and Activities of Daily Living (ADLs).

A clinical and functional assessment with motion capture was carried out. The motion data was collected using the Motion Analysis® optical motion capture system [113] with fourteen Raptor-E optoelectronic cameras sampling at 100 Hz. The subjects agreed to wear a motion capture suit, attached with a set of reflective markers on their body based on Helen Hayes marker set structure [114]. The output of the motion capture system is a set of 3D marker's positions in the temporal domain.

Adapting the protocol from [115], all subjects were asked to walk naturally along a 10 meter walkway with their normal gait speed. The length is bounded by the capturing volume of the motion capture system. Four trials of walking were performed and a rest period of two minutes was given after each trial. The first trial was for practising and the last was for cooling down. We only consider the second and third trials as they better represent the subjects' normal walking motions.

3.4 Skeleton-Based Feature Extraction

In this section, we explain how we process the data and extract the features from a skeleton-based motion format.

Please fill in the blanks or mark in the blanks that most correspond to you.

1. Name .....
2. Gender .....
3. Age ..... years
4. Weight ..... KG
5. Height..... CM
6. Academic degree .....
7. Do you have a regular disease? (1) No.
(2) Yes. .....
8. Do you have bone and muscle problems like pain in the neck? (1) No.
(2) Yes. .....
9. Do you have a neurological problem like head injury or stroke? (1) No.
(2) Yes. .....
10. Are you taking medication currently? (1) No.
(2) Yes. Specify the name of the medicine: .....
11. Do you have problems with your eyes, such as blurry eyes? (1) No.
(2) Yes. The problem was solved
by .....
12. Are you exercising regularly? (1) No.
(2) Yes. (More than or equal to 3 times/week)
13. In the last 24 hours, have you been drinking alcohol? (1) No. (2) Yes.
14. In the last 12 months, have you ever fallen?
(1). No. (2) Yes.
15. Time Up and Go Test (TUG) ..... sec
16. Five times Sit to Stand Test ..... sec
17. Thai Mini-Mental State Examination (TMSE) Score .....
18. Thai Fall Efficacy Scale International .....
19. Can a minimum of 8 meters be reached?
(1) No. (2) Yes.
20. Can you understand and communicate in Thai?
(1) No. (2) Yes.

Fig. 3.2 Questionnaire used in the interviews.

3.4.1 Data Preprocessing

The data from the motion capture system is expressed in a 3D marker position format. Such a format is inefficient for motion classification as it depends heavily on the body size and phenotype. Following existing methods on motion analysis such as [116], we convert the marker position based format into a skeletal format, such that we can focus on the movement of joints instead of body surfaces.

The conversion to a skeletal format is facilitated by inverse kinematics [117], a process that calculates the angle of a skeletal joint from multiple markers on the body surface. Motion retargeting is performed to obtain a skeleton with predefined dimensions. In our system, all these functionalities are provided by the software Autodesk MotionBuilder.

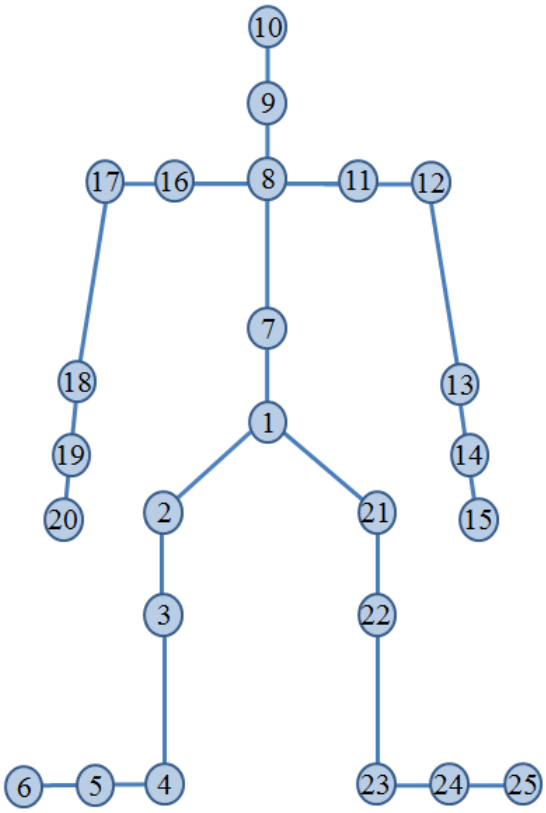


Fig. 3.3 The skeleton structure

The skeletal definition we adopted consists of 25 joints corresponding to real-human joint positions, as shown in Fig. 3.3. Fig. 3.4 shows sampled frame of the skeleton in a gait cycle. We store the skeletal joint rotation data in the BioVision (BVH) format, which is a popular format readable by a large number of software libraries [118]. Notice that the end

Fig. 3.4 Sampled keyframes in one walking cycle.

effectors do not contain any rotation information as they do not have child joints, and they are excluded in the feature extraction process.

We normalize the motion data of the walking cycles in the spatial domain by aligning the first frame to 0 degree, such that the gait classification process is not affected by the initial walking direction. Furthermore, to normalize the temporal variation, we perform dynamic time warping (DTW) [119] to wrap all captured motion into the mean duration of the walking cycles, such that the classifier is not affected by the duration of the motion. These processes are supported by the software MATLAB® version 2016b.

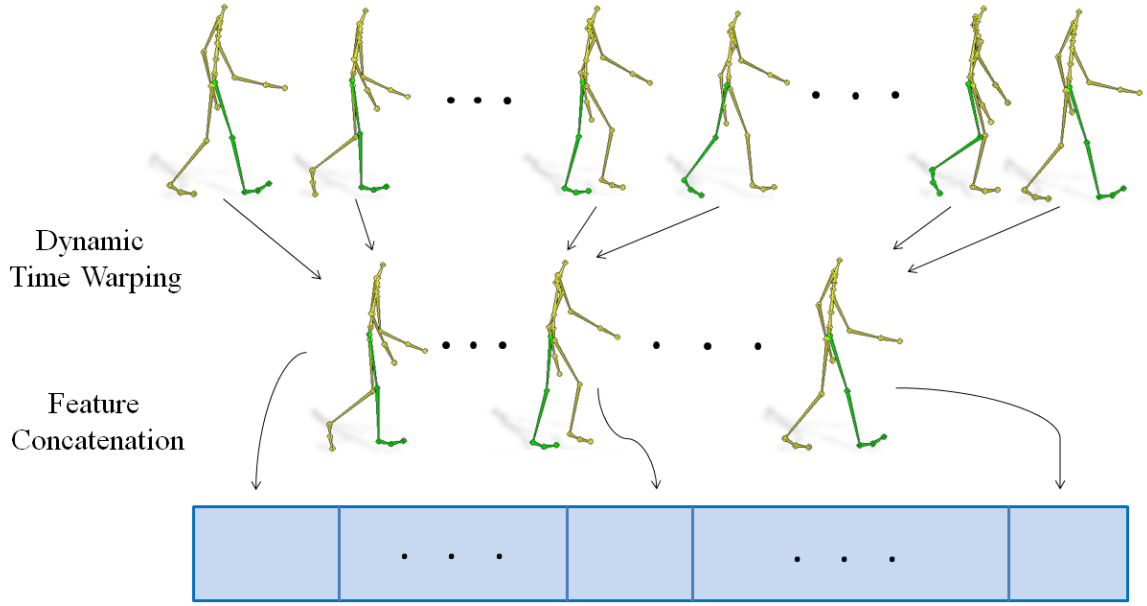


Fig. 3.5 The extraction of the feature vector.

3.4.2 Feature Extraction

In the two sections below, we explain how the proposed features based on relative joint information, known as 3DRJDP and 6DSymRJDP, are extracted. We also describe other traditional features, for which the features are extracted from individual joints.

The considered features below are defined for each frame of the motion. In order to evaluate the continuous sequence of motion features over time, we concatenate the frame-based features over time into a long feature vector. This process is visualized in Fig. 3.5.

In the following explanation, i and j represents joints. (x_i, y_i, z_i) represents the 3D position of joint i, and (x_j, y_j, z_j) represents that of joint j. φ_i, θ_i, ω_i are the roll, pitch and yaw angles of joint i respectively. $(x_{hips}, y_{hips}, z_{hips})$ represents the 3D position of the hips joint. n is the total number of joints.

3.4.3 The Proposed Features

3D Relative Joint Displacement (3DRJDP) is defined as the displacement between all the possible joint pairs in the skeletal hierarchy, excluding the pairs connecting to the same joint. We extract relative joint displacement as:

$$D_{3DRJDP}(i,j) = \left\{(x_i - x_j),\ \ (y_i - y_j),\ \ (z_i - z_j)\right\}. \tag{3.1}$$

3DRJDP within one frame consists of $n(n-1)$ items:

$$\begin{aligned}&\{D_{3DRJDP}(1,2),\ \ D_{3DRJDP}(1,3),\ \ D_{3DRJDP}(1,4),\ \ ...\\ &D_{3DRJDP}(2,1),\ \ D_{3DRJDP}(2,3),\ \ ...\ \ D_{3DRJDP}(n,n-1)\}.\end{aligned} \tag{3.2}$$

6D Symmetric Relative Joint Displacement (6DSymRJDP) is defined as the pair-wise displacement between all the possible joint pairs, excluding the pairs connecting to the same joint. Also, because of the symmetric nature of the feature, we also exclude the pairs with joint identifier $i > j$:

$$\begin{aligned}D_{6DSymRJDP}(i,j) = \ &\left\{(x_i - x_j),\ \ (y_i - y_j),\ \ (z_i - z_j),\right.\\ &\left.(x_j - x_i),\ \ (y_j - y_i),\ \ (z_j - z_i)\right\}.\end{aligned} \tag{3.3}$$

6DSymRJDP within one frame consists of $\frac{n(n-1)}{2}$ items:

$$\begin{aligned}&\left\{D_{6DSymRJDP}(1,2),\ \ D_{6DSymRJDP}(1,3),\right.\\ &D_{6DSymRJDP}(1,4)\ \ ...\ \ D_{6DSymRJDP}(2,3)\ \ ...\\ &\left.D_{6DSymRJDP}(n-1,n)\right\}.\end{aligned} \tag{3.4}$$

3.4.4 Existing Features Considered in this Work

3D Joint Angle (3DJA) is defined as the concatenation of Euler joint rotation angle for each joint:

$$D_{3DJA}(i) = \{\varphi_i,\ \theta_i,\ \omega_i\}. \tag{3.5}$$

3DJA within one frame is therefore:

$$\{D_{3DJA}(1),\ D_{3DJA}(2),\ \ldots\ D_{3DJA}(n)\}. \tag{3.6}$$

4D Joint Angle (4DJA) is extracted by converting the Euler 3DJA rotation into the quaternion representation. The quaternion representation avoids problems such as ambiguity and gimbal locks in the Euler angles system:

$$D_{4DJA}(i) = \{w_i,\ a_i,\ b_i,\ c_i\}, \tag{3.7}$$

where the quaternion parameters are calculated as: 4DJA within one frame is therefore:

$$\{D_{4DJA}(1),\ D_{4DJA}(2),\ \ldots\ D_{4DJA}(n)\}. \tag{3.8}$$

3D Relative Joint Distance (3DRJD) is defined as the concatenation of the Euclidean distance in each dimension between the all possible joints pairs, except the self-connecting pairs and the neighboring joint pairs as the distances of these pairs are constant:

$$D_{3DRJD}(i,j) = \left\{\sqrt{(x_i - x_j)^2},\ \sqrt{(y_i - y_j)^2},\ \sqrt{(z_i - z_j)^2}\right\}. \tag{3.9}$$

3DRJD within one frame is therefore:

$$\begin{aligned}&\{D_{3DRJD}(1,2),\ D_{3DRJD}(1,3),\ D_{3DRJD}(1,4)\\ &\ldots\ D_{3DRJD}(n-1,n)\}.\end{aligned} \tag{3.10}$$

1D Relative Joint Distance (1DRJD) is the 1D version of 3DRJD by combining the 3D distance into a 1D distance:

$$D_{1DRJD}(i,j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}. \tag{3.11}$$

1DRJD within one frame is therefore:

$$\begin{aligned}&\{D_{1DRJD}(1,2),\ D_{1DRJD}(1,3),\ D_{1DRJD}(1,4)\\ &\ldots\ D_{1DRJD}(n-1,n)\}.\end{aligned} \tag{3.12}$$

3D Hips Relative Joint Position (3DhipRJP) is defined as the concatenation of the Euclidean distance of each dimension of all joints with respect to the hip position, except the joint itself and the neighboring joints of the hips, as they are constant distances from the hips:

$$\begin{aligned} &D_{3DhipRJP}(i) \\ &= \left\{ \sqrt{(x_{hips} - x_i)^2},\ \sqrt{(y_{hips} - y_i)^2},\ \sqrt{(z_{hips} - z_i)^2)} \right\}. \end{aligned} \tag{3.13}$$

3DhipRJP within one frame is therefore:

$$\left\{ D_{3DhipRJP}(1),\ D_{3DhipRJP}(2),\ \ldots\ D_{3DhipRJP}(n) \right\}. \tag{3.14}$$

1D Hips Relative Joint Position (1DhipRJP) is the 1D version of 3DhipRJP by combining the 3D distance into a 1D distance:

$$\begin{aligned} &D_{1DhipRJP}(i) \\ &= \sqrt{(x_{hips} - x_i)^2 + (y_{hips} - y_i)^2 + (z_{hips} - z_i)^2}. \end{aligned} \tag{3.15}$$

1DhipRJP within one frame is therefore:

$$\left\{ D_{1DhipRJP}(1),\ D_{1DhipRJP}(2),\ \ldots\ D_{1DhipRJP}(n) \right\}. \tag{3.16}$$

3.5 Feature Selection Algorithms

To extract the useful information from the raw data in a lower dimensional format, we apply feature selection algorithms before doing classification. These algorithms are independent of the input feature types. Given a type of input feature (e.g. 3DJA, 3DRJD, 3DRJDP, 6DSymRJDP), these algorithms obtain a subset of the features to be used in the classification algorithm in the next stage. Using 3DJA as an example, one possible solution from these algorithms could consider only the lower body joint angles but discard the upper body ones. The underlying motivation is that some dimensions of the features are more relevant to the classification problem, while some may either be irrelevant or noisy. By selecting only the feature subset that is helpful for classification, the size of the input dataset can be reduced and the classification accuracy can be improved.

In this research, we employ three algorithms to measure the discriminativeness of each feature and select the optimal subset, including (1) F-score [25], (2) Neighborhood Compo-

nent Analysis (NCA) [27], and (3) ReliefF [26]. These methods have achieved great success in other problems, and we evaluate their performance in human motion analysis.

3.5.1 F-score

Here, we explain our implementation of F-score, which measures the discrimination power within a set of training data with predefined criterion functions to characterize the intrinsic properties of the training data.

Given the training vectors p_k, $k = 1,...,m$, and the number of members in each of the C different gait classes (e.g. healthy, muscle weakness) as n_c, the F-score of the l^{th} feature is defined as:

$$w_l = \sum_{c=1}^{C} (\overline{p}_l^{(c)} - \overline{p}_l)^2 / \sum_{c=1}^{C} \frac{\sum_{k=1}^{n_c} (p_{k,l}^{(c)} - \overline{p}_l^{(c)})^2}{n_c - 1} \tag{3.17}$$

where C is the total number of gait classes, $\overline{p}_l$ and $\overline{p}_l^{(c)}$ are the average values of the whole dataset and the gait class c respectively, $p_{k,l}^{(c)}$ is the l^{th} feature of the k^{th} instance in the gait class c.

The numerator in (3.17) represents the discrimination among all category sets, and the denominator represents the discrimination within each of the sets. Since more discriminative features are represented by larger F-score values, we use the scores to rank the importance of the joints features and select the best I features that maximize the classification accuracy.

3.5.2 Neighborhood Component Analysis (NCA)

Neighborhood Component Analysis (NCA) is a dimensional reduction method that improves the predicting performance of the K-Nearest Neighbor (KNN) classifier being used in the feature selection process.

In [120], NCA has been proposed as the method to learn a Mahalanobis distance measurement for maximizing a stochastic variant of the leave-one-out cross-validation within KNN in the training dataset. This Mahalanobis distance can be calculated using inverse square roots and represented as symmetric positive semi-definite matrices. Here, the probability of a training vector, p_i, selecting another training vector, p_j, as its reference point is defined as:

$$P_{ij} = \begin{cases} \dfrac{K(D_w(p_i, p_j))}{\sum_{k \neq i} D_w(p_i, p_k)} & \text{if} \quad i \neq j, \\ 0 & \text{if} \quad i = j, \end{cases} \tag{3.18}$$

where $K(\Delta) = exp(-\Delta/\sigma)$ is a kernel function, the kernel width σ is an input parameter that influences the probability of each point being selected as the reference point, w is a weighting vector, and D_w is the weighted distance between two samples.

The objective function of KNN with the approximate leave-one-out classification accuracy can be written as:

$$\xi(w) = \frac{1}{Q}\sum_i \sum_j y_{ij} P_{ij}, \tag{3.19}$$

where $y_{ij} = 1$ if and only if $y_i = y_j$ and $y_{ij} = 0$ otherwise. Q is the total number of samples.

We follow [27] to adopt the NCA strategy into a feature selection task by including a weighting score, which results in a nearest neighbor-based feature weighting methods for reducing the high dimensionality of the input vector. Such a method modifies the original KNN and improves the classification performance in the leave-one-out cross-validation method. Here, Eq. 3.19 is modified for approximating the leave-one-out classification accuracy in KNN as:

$$\xi(w) = \sum_i \sum_j y_{ij} P_{ij} - \lambda \sum_{l=1}^{d} w_l^2, \tag{3.20}$$

where $\lambda > 0$ is a regularization parameter and can be tuned via cross-validation, which replaces the functionality of the coefficient $\frac{1}{Q}$ in Eq. 3.19, w_l is the scores that defines the ranking of the features, and it is obtained by gradient decent method. Essentially, by using the feature weighting method within the context of KNN, we identify the extent of redundancy in the features and select the optimal feature subset for better classification.

3.5.3 ReliefF

The original RELIEF algorithm [121] is used to evaluate the features of the data samples, resulting in a value that distinguishes a sample from it neighbors. The quality of the features is represented by its weight, which is estimated from two neighbor samples, including one nearest value from the same class (i.e. nearest hit) and another nearest value from a different class (i.e. nearest miss). The weighting score vector of the feature l on the sample r, $w_l(r)$, is estimated using the normalization of all Q training samples with the following equation:

$$w_l(r) = w_l(r) - \frac{\text{diff}(l,r,h)}{Q} + \frac{\text{diff}(l,r,m)}{Q}, \tag{3.21}$$

where r is the feature value of the considering sample, h is the feature value of the nearest hit sample, and m is the feature value of the nearest miss sample, $\text{diff}(l,r,h)$ calculates the distance between samples r and h using the l^{th} feature, $\text{diff}(l,r,m)$ calculates that between

samples r and m. However, RELIEF is limited to only two-class problems and is sensitive to noisy data samples.

Therefore, ReliefF [122] is proposed to improve the reliability in estimating the weight and has been extended to handle multi-class data sets while retaining the same computational complexity. Here, the K-nearest neighbor (KNN) concept is adopted to find the k nearest hit observations and k nearest miss observations, instead of using one nearest neighbor in each class as in Eq. 3.21, which improves the reliability of weight approximation. A multiple-class problem is formulated as finding one nearest miss, $M(\Phi)$, for each different class Φ and averaging their contribution for updating the estimated w_l:

$$\begin{aligned} w_l(r) = w_l(r) - & \frac{\text{diff}(l,r,h)}{Q} \\ & + \sum_{\Phi \neq class(r)} \frac{\frac{P(\Phi)}{1-P(class(r))} \times \text{diff}(l,r,M(\Phi))}{Q}, \end{aligned} \tag{3.22}$$

The final weighting score of the feature, w_l is the summation of difference among the sample and its neighbors from all Q samples. It estimates the ability of features to separate different classes.

3.6 Motion Classification

In supervised learning-based classification problems, the Support Vector Machine (SVM) [14] is a powerful classifier that is commonly used to classify different categories of the data. It maps the data to a higher dimensional space and separates the data using hyperplanes. Such hyperplanes are optimal boundaries that categorise new sets of data. The optimised hyperplanes would maximise the margin among classes in the training data.

Given an instance-label pair (p_i, q_i), $i = 1, \ldots, l$ where p_i is a sample vector and $q_i \in \{1, -1\}^l$ represent one of the two the categories in the training data, the original SVM [123, 124] obtains the solution as an optimization problem as follow:

$$\begin{aligned} \min_{w,b,\xi} \quad & \frac{1}{2} w^T w + C \sum_{i=1}^{l} \xi_i \\ \text{subject to} \quad & q_i (w^T \phi(p_i) + b) \geq 1 - \xi_i, \\ & \xi_i \geq 0. \end{aligned} \tag{3.23}$$

where the feature vectors p_i are mapped onto the higher dimensional space using the function ϕ, $C > 0$ is the penalty parameter for the error. w is known as the weight vector, b is the

bias and ξ is the maximum margin. In Eq. 3.23, the idea is to apply a linear hyperplane to identify the margin of each data class as shown in Fig. 3.6 left.

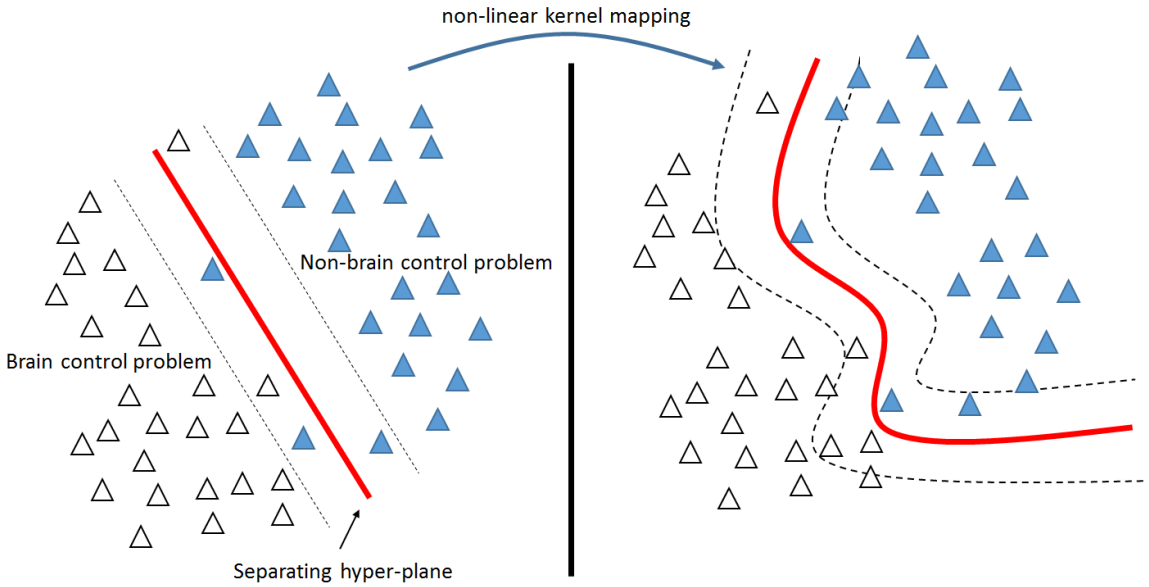


Fig. 3.6 Support Vector Machine classifiers.

However, in a multiple classes problem, the linear SVM with one hyperplane cannot separate multiple data classes. As a solution, the *one-against-one* approach is proposed [125, 126], in which for a M classes classification problem, $M(M-1)/2$ classifiers are constructed and each of them trains data from two classes. Given training data from the i^{th} and the j^{th} classes, the solution can be found by the following objective function:

$$\begin{aligned} \min_{w^{ij},b^{ij},\xi^{ij}} \quad & \frac{1}{2}(w^{ij})^T w^{ij} + C\sum_t (\xi^{ij})_t \\ \text{subject to} \quad & (w^{ij})^T \phi(p_t) + b^{ij} \geq 1 - \xi_t^{ij}, \text{if } p_t \text{ in the } i^{th} \text{ class}, \\ & (w^{ij})^T \phi(p_t) + b^{ij} \leq -1 + \xi_t^{ij}, \text{if } p_t \text{ in the } j^{th} \text{ class}, \\ & \xi_t^{ij} \geq 0. \end{aligned} \tag{3.24}$$

where w^{ij} is the weighting vector, b^{ij} is the bias and ξ^{ij} is the maximum margin. A voting strategy is implemented to find the class label that fits the best to the testing data. In particular, the class label that has a maximum number of votes from all binary classifiers is considered to be the best fit class label.

In a complex data space, it may be difficult to apply linear hyperplanes to separate the classes. Therefore, kernel functions, $K(p_i, p_j) \equiv \phi(p_i)^T \phi(p_j)$, are proposed to construct

Feature Types	Muscle Weakness	Joint Problem	Neurological Defect	Healthy	p-value	Baseline Classification
3DJA	-0.2 ± 3.08	0.2 ± 4.93	0.89 ± 3.47	0.36 ± 3.86	0.871	57.14
4DJA	-1.2 ± 0.54	-0.3 ± 0.84	-0.4 ± 0.93	-0.4 ± 0.34	0.043	51.43
1DRJD	60.6 ± 1.63	62.5 ± 2.63	64.2 ± 2.58	62.5 ± 2.94	0.578	55.86
3DRJD	16.7 ± 6.73	12.6 ± 2.61	13.8 ± 3.21	15.7 ± 5.68	**0.001**	61.43
1DhipRJP	50.9 ± 0.76	51.7 ± 0.98	51.6 ± 0.98	49.2 ± 0.67	0.168	50.82
3DhipRJP	-3.9 ± 1.98	-6.1 ± 3.75	-5.2 ± 3.67	-5.4 ± 2.11	0.389	57.14
3DRJDP	17.2 ± 6.81	9.7 ± 2.65	12.8 ± 4.21	11.7 ± 3.78	**0.001**	63.29
6DSymRJDP	17.2 ± 6.81	9.7 ± 2.65	12.8 ± 4.21	11.7 ± 3.78	**0.001**	**67.14**

Table 3.2 The numerical statistics of the extracted features. p-value and baseline classification referred to the primary setup.

higher dimensional hyperplanes, as shown in Fig. 3.6. In this study, we evaluate three popular kernel functions including linear, polynomial, and radial basis function (RBF) as defined below:

- Linear:

$$K(p_i, p_j) = p_i^T p_j \tag{3.25}$$

- Polynomial:

$$K(p_i, p_j) = (\gamma p_i^T p_j + r)^d, \quad \gamma > 0 \tag{3.26}$$

- Radial basis function (RBF):

$$K(p_i, p_j) = exp(-\gamma ||p_i - p_j||^2), \quad \gamma > 0 \tag{3.27}$$

where, γ, r, and d are kernel parameters that need to be optimized during the construction of the model. In this research, we implemented a multi-class SVM using the library LIBSVM [126]. We conducted a grid-search to tune the kernel parameters.

3.7 Experimental Results

Using the database constructed in Section 3.3, we evaluate the performance of our proposed method. Our *primary experiment setup* utilizes only the three unhealthy classes for classification, while our *secondary setup* utilizes all four classes, including the healthy class. The former is considered practically important to service providers, as the majority of patients only access health-care services when they are unhealthy. The latter includes a healthy control group in order to verify the performance of the system.

3.7.1 Evaluation on Different Kinematics Features

Table 3.2 summarises the numerical statistics of all features extracted from the participants' motion. For example, for 6DSymRJDP, the average value of the features is 17.2 ± 6.81 millimetre in the muscle weakness category. Notice that 3DRJDP and 6DSymRJDP have the same statistics because of the similar ways the features are defined, but the former has more items in the feature vector, which affects the performance of classification.

We analyse the values of the features using ANOVA to identify the significance of variance using the primary setup. We define p-value <= 0.05 as an indication of significant variance, showing that the features can be used to differentiate different types of gait disorders.

The results are shown in the fifth column of Table 3.2. It can be observed that features such as 3DJA and 4DJA have high p-values, meaning that there is not sufficient evidence to reject the null hypothesis. The three better features for gait classifications are 6DSymRJDP, 3DRJDP, and 3DRJD.

Next, we test the performance of the features by using the whole feature vectors for SVM classification. For each type of feature, we experiment with different SVM kernels including linear, polynomial and radial basis function (RBF), and select the best classification result. The accuracy is obtained using the leave-one-out cross-validation strategy. As shown in the last column of Table 3.2, both of our proposed features, 3DRJDP and 6DSymRJDP, outperform the other features. In particular, 6DSymRJDP performs the best by delivering an accuracy of 67.14%. This demonstrates that the relationship between joint pairs carries more information for gait classification comparing to the absolute values of individual joint features. Comparing with 3DRJDP, 6DSymRJDP has only the half number of items in the feature vector by pairing up logically relevant ones, making it easier for the SVM systems to model the data and interpret the diagnostic results.

3.7.2 Evaluation on Different Feature Selection Methods

We compare different features under four different feature selection strategies: the baseline method without any feature selection, F-score, NCA and RelieF. The selected features are concatenated as the feature vector for representing each motion based on the selected key frames. They are fed as an input vector into the SVM machine learning mechanism. For each type of feature, we experiment with different numbers of features and obtain the optimal value. We also experiment with different kernels including linear, polynomial and RBF, and use the value from the best one.

As shown in Table 3.3, generally, the results with feature selection outperform the baseline ones. Also, the best results are obtained by using the F-score feature selection method. Among eight different features, 6DSymRJDP and 3DRJDP perform better than the other features under most feature selection methods. The highest accuracy 84.29% (primary setup) and 79.17% (secondary setup) appears when we use 6DSymRJDP under F-score feature selection method. With NCA, the best accuracy achieved is 75.71% (primary setup) and 72.81% (secondary setup) with the 3DJA feature type, while our proposed feature 6DSymRJDP achieves comparable values. With ReliefF, by using the default value of $k = 10$ into KNN [122], the best classification performance is 80.00% of accuracy on both 1DRJD and 6DSymRJDP (primary setup), and 74.56% on ReliefF (secondary setup).

F-score utilizes the whole training set in evaluating the suitability of a feature in classification, while NCA and ReliefF consider different sub-parts of the training set under the

Features	Baseline	F-score	NCA	ReliefF
3DJA	57.14	83.17	75.71	78.57
4DJA	51.43	51.43	52.86	72.86
1DRJD	55.86	78.57	71.43	80.00
3DRJD	61.43	81.43	71.43	65.71
1DhipRJP	50.82	52.86	51.43	48.29
3DhipRJP	57.14	64.29	60.00	62.86
3DRJDP	63.29	82.43	72.86	76.86
6DSymRJDP	67.14	**84.29**	74.82	80.00

Features	Baseline	F-score	NCA	ReliefF
3DJA	49.71	78.61	72.81	74.56
4DJA	46.32	47.82	45.67	68.43
1DRJD	47.78	74.54	68.31	71.34
3DRJD	55.16	75.43	72.56	58.43
1DhipRJP	41.56	44.65	47.32	51.71
3DhipRJP	48.91	59.71	53.45	57.43
3DRJDP	57.76	77.33	65.89	71.47
6DSymRJDP	60.17	**79.17**	69.78	72.43

Table 3.3 Classification accuracy of each feature using different feature selection methods for (upper) the primary setup (lower) the secondary setup.

Features	Linear	Polynomial	RBF
3DJA	74.29	83.17	51.43
3DRJD	81.43	80.00	55.71
3DRJDP	76.57	82.43	59.17
6DSymRJDP	81.43	**84.29**	62.86

Features	Linear	Polynomial	RBF
3DJA	70.15	78.61	50.65
3DRJD	77.64	75.43	46.34
3DRJDP	74.14	77.33	54.23
6DSymRJDP	78.45	**79.17**	56.46

Table 3.4 The performance of SVM kernels with F-score for (upper) the primary setup (lower) the secondary setup.

KNN algorithm. Since many databases in this field are small, F-score has an advantage of utilizing the full dataset for a more robust performance. This explains why it outperforms other feature selection methods in our experiments.

3.7.3 Kernel and Classifier Analysis

We evaluate the kernel selection process using F-score as the feature selection algorithm, as it performed best in the previous experiment. We evaluate the features that perform well with F-score (>80%), including 3DJA, 3DRJD, and our prposed features 3DRJDP and 6DSymRJDP.

Table 3.4 shows the experiment of all kernels. The polynomial kernel generates the best results with 6DSymRJDP using F-score, with the classification accuracy of 84.29% (primary setup) and 79.17% (secondary setup). In general, the polynomial kernel performs better than the linear and RBF kernels, showing that the polynomial kernel models the motion features better.

Fig. 3.7 shows the classification accuracy against the number of features selected using F-score with 6DSymRJDP using the primary setup. We see that selecting more features may not necessarily improve the classification accuracy. The classification accuracy starts to drop when the number of features exceeds the modelling power of the classification system. The best result is obtained with the polynomial kernels with 48 features selected, indicated by the orange line.

To evaluate different classifiers, we compare the performance of SVM (using the polynomial kernel), neural network and binary decision tree. The neural network is implemented

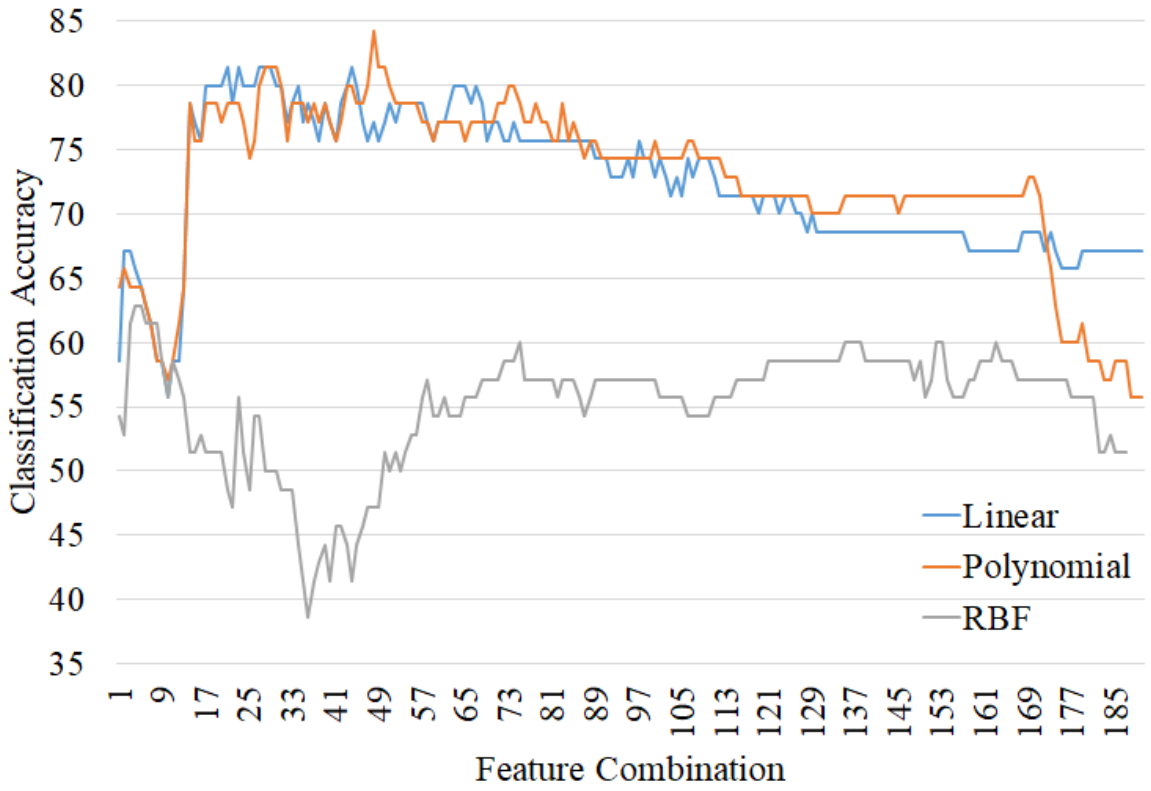


Fig. 3.7 Feature selections according to different kernels for 6DSymRJDP with F-score.

Features	SVM	Neural Network	Decision Tree
3DJA	83.17	51.32	79.19
3DRJD	81.43	54.89	77.52
3DRJDP	82.43	53.13	79.63
6DSymRJDP	**84.29**	56.87	81.24

Features	SVM	Neural Network	Decision Tree
3DJA	78.61	45.71	71.15
3DRJD	75.43	48.95	73.34
3DRJDP	77.33	50.67	72.65
6DSymRJDP	**79.17**	52.57	74.33

Table 3.5 The performance of different classifiers with F-score for (upper) the primary setup (lower) the secondary setup.

using a 4-layer structure. The neurons number in the input (i.e. first) layer and the output (i.e. last) layer is equal to the feature dimensionality and the number of classes respectively. The second layer has half the neurons of the first layer, and the third layer has half the neurons of the second layer, thereby implementing a typical triangle structure. The sigmoid function is used as the activation function. The binary decision tree does not require any particular parameter setup. Table 3.5 shows the results and SVM achieves the best accuracy of 84.29% (primary setup) and 79.17% (secondary setup). Neural networks typically require a larger amount of training data, and therefore perform sub-optimally on smaller clinical databases. Decision trees have weaker generalization power comparing to polynomial SVM, as they only utilize planer decision boundaries.

The gender bias in our database is unfortunately unpreventable due to the local culture. In Thailand where the data is collected, females have much stronger local social networks comparing to males. These networks are more open to voluntary works including ours - working with technicians to capture motion data. In our current database, we do not have sufficient data to evaluate if the gender bias has affected the experimental accuracy. One future direction is to analyze if (1) there is a correlation between genders and the types of gait disorder, and (2) the gender ratio would affect the performance of the proposed framework.

3.7.4 Conclusions

In this chapter, we propose an automatic gait analysis frame- work for musculoskeletal and neurological disorder diagnosis. We capture the gait motion from 45 Thai people with ages ranging between 61 and 91 according to four disorder categories: healthy, muscle weakness, joint problem and neurological defect. After that, we map all the gait motion into fixed length sequence with dynamic time warping and represent the warped gait sequences with different gait features. We experiment with several combinations of the optimal joint set, feature extraction methods, and classifiers for the disorder diagnosis. The results show that using our proposed feature 6D Symmetric Relative Joint Displacement (6DSymRJDP) can achieve better results (84.29%) than using other gait features.

Since it is inconvenient and uncomfortable for the older people to wear suits and markers for motion capture, one future direction is to explore R-GBD motion sensing hardware such as the Microsoft Kinect, which does not require capture suits nor calibration. A benefit is that such a type of motion capture equipment can be deployed in care homes for everyday motion tracking and pre-diagnosis, thereby helping to identify disorders in the early stage.

Also, while our system allows the machine to automatically extract features for disorders classification, it may be enhanced by introducing prior medical knowledge to the existing

features. We envisage that a combination of human knowledge and machine understanding would give a better description on these problems.

While there can be overlapping between different classes of disorder, when selecting subjects for motion capture, we found that there were not enough subjects suffering from more than one disorder to form a representative class. Therefore, we did not include those subjects. One of our future directions is to gather enough data of subjects suffering from multiple types of disorder and design a corresponding classification system. With enough data, one possible idea is to implement one binary classifier for each disorder to tell if the patient is suffering from such a disorder or not. Multiple binary classifiers are then combined to form the full system.

Chapter 4

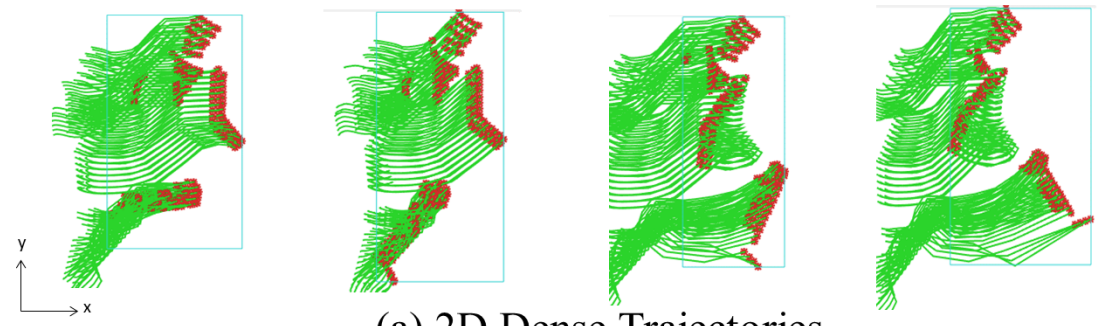
Learning Arbitrary view features for action recognition

Unlike 3D skeletal data, the appearances of a particular 2D action video from different viewpoints vary dramatically, which results in dissimilar visual features. Conventional arbitrary view action recognition methods usually leverage the view-invariance of 3D models, which may cause results highly dependent on the quality of the 3D model. However, the 3D models calibrated from different viewpoints are always unrealistic, and the transfer process of view-invariance is not smooth.

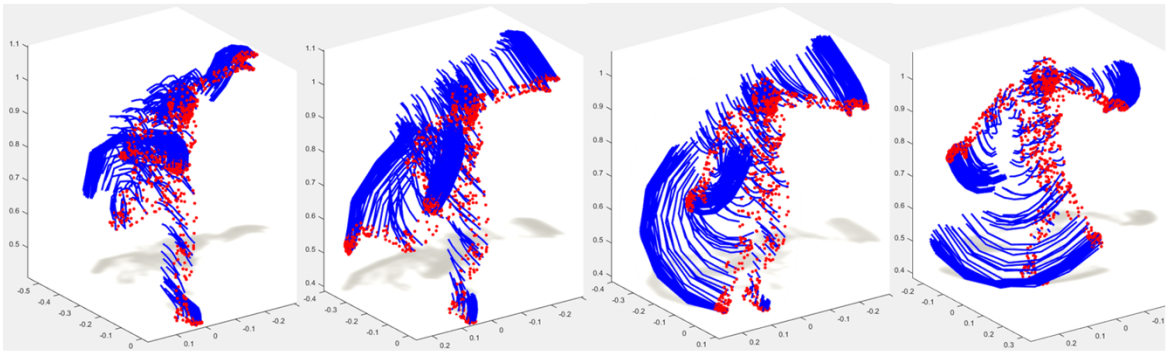
Therefore, in this chapter, we adopt 3D models built with computer graphics technology to assist in solving the problem of arbitrary view action recognition. As a solution, a new transfer dictionary learning framework that utilises computer graphics technologies to synthesise realistic 2D and 3D training videos is proposed, which can project a real-world 2D video into a view-invariant sparse representation. A new 3D feature set called the 3D dense trajectories consisting of 3D trajectories, *3D Histogram of Optical Flow* (3DHOF) and *3D Motion Boundary Histogram* (3DMBH) is also proposed for a better description of synthesised motion in 3D.

4.1 Introduction

Most of the existing works leverage view-invariance provided by 3D models to realize cross-view or arbitrary-view action recognition. Traditionally, simplified cylindrical models are used [1, 2], which does not generate realistic movement appearance. High-quality reconstruction models are proposed by calculating them from multi-view 2D videos [4]. In order to increase the system robustness to viewpoint changes, training data is forced to cover



(a) 2D Dense Trajectories



(b) Proposed 3D Dense Trajectories

Fig. 4.1 Leveraging view-invariance from 3D model is a popular idea to tackle arbitrary-view and cross-view action recognition. (a) Existing works [1, 2] project a simplified 3D cylindrical model into as many viewpoints as possible to produce 2D training videos and extract *2D dense trajectories* from these projections. However, some human appearance information could be lost due to the unrealistic 3D reconstruction and the simplified cylindrical model. The discrete projection angles also inevitably result in the loss of 3D geometric information. (b) The proposed *3D dense trajectories* are extracted directly from high-quality 3D human surface model without any projection.

as much 2D data projected along as many viewpoints as possible. All these approaches suffer from the following problems: (1) The recognition accuracy is highly related to the quality of 3D models. Some human appearance information could be lost due to the unrealistic 3D reconstruction and the simplified cylindrical model; (2) Despite the effort to project the 3D model into as many viewpoints as possible, these discrete projection angles will inevitably result in the loss of 3D geometric information. A large amount of 2D projections also requires larger system capacity and training cost.

To solve the problems, we synthesize training data using high-quality human models with captured 3D motion data. We employ primary deformation [127] to drive the movement of the models, and motion retargeting [128] to adjust the movement based on the body sizes of the models. We further propose a new 3D feature set called the *3D dense trajectories* including 3D trajectories, 3DHOF and 3DMBH. This allows us to extract the feature directly from 3D videos and avoid geometric information loss due to discrete projection. Finally, we propose a new view-invariance transfer dictionary learning framework, which extracts the view-invariance between 3D and 2D video, to perform arbitrary view action recognition. We pre-train the system with a large number of automatically synthesized 3D and 2D videos. This allows us to train a view-invariant action classifier using only a small number of real-world 2D videos, in which the view information is not annotated. Experimental results show that our system achieves better accuracy when compared with previous work in arbitrary-view and cross-view action recognition.

This chapter has three main contributions:

- We propose a new transfer dictionary learning framework that utilizes synthetic 2D and 3D training videos generated from realistic human models to learn a dictionary that can project a real world 2D video into a view-invariant sparse representation, which allows us to train an action classifier that works in an arbitrary view.

- We release our synthetic 2D and 3D dataset for public usage. This is the first structured action dataset built with realistic human models for high-quality action classification.

- We propose a new 3D feature set called the 3D dense trajectories consisting of 3D trajectories, 3DHOF and 3DMBH for a better description of motion in 3D. This can be considered as a 3D counterpart of the popular 2D feature dense trajectories [51].

This chapter is based on our previous work presented in [3], but it substantially extends the work in four aspects, which are: (1) We replace the cylinder-based 3D model with several more realistic 3D human models. The motion is retargeted according to the bone dimensions [128] and skinned to the realistic models [127]. (2) We propose the 3D dense trajectories

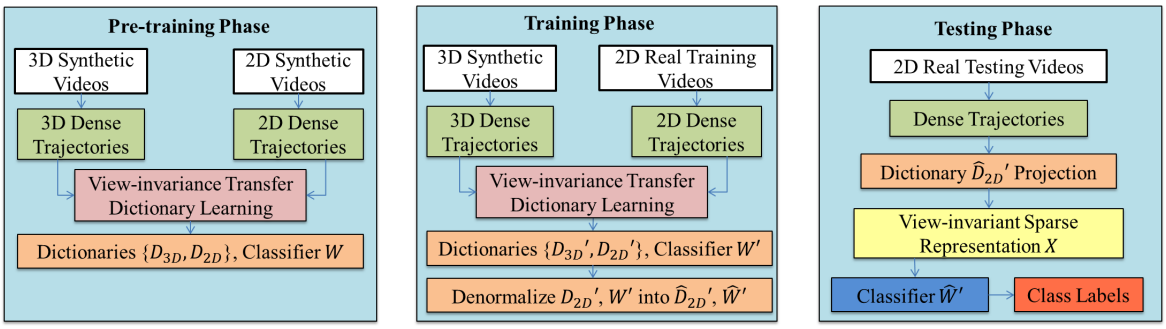


Fig. 4.2 The overview of our view-invariance transfer dictionary learning system. (Left) In the pre-training phase, we learn the dictionaries D_{3D}, D_{2D} and a linear classifier W simultaneously from the synthetic 3D videos and the synthetic 2D videos. (Middle) In the training phase, we replace the synthetic 2D videos with 2D real training videos for adapting the dictionaries D_{3D}', D_{2D}' and the classifier W'. The 2D dictionary and the classifier are denormalized into $\widehat{D}_{2D}'$ and $\widehat{W}'$ respectively. (Right) In the testing phase, given any real 2D video, we apply $\widehat{D}_{2D}'$ to encode the features into a view-invariant sparse representation X, and use $\widehat{W}'$ for classification.

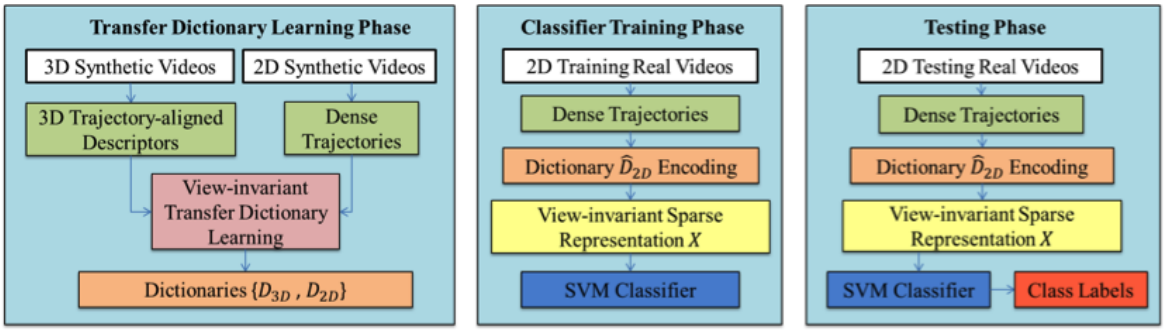


Fig. 4.3 Overview of our baseline view-invariant human action recognition system in [3]. The major weakness of this framework is this is not an end-to-end system. A separate SVM classifier is required to recognise the action.

including 3D trajectories, 3DHOF and 3DMBH to better describe the motion in 3D videos. (3) By jointly training the transfer dictionary pair and the classifier, we build an end-to-end framework with an updated objective function to improve the efficiency and performance of the system. (4) We perform more detailed system evaluation with two more datasets: i3DPost and UWA3DII.

The rest of this chapter is organized as follows. In Section 4.2, we give an overview of our view-invariant human action recognition frame. In Section 4.3, we present the synthesis and feature extraction process on our 2D and 3D video data. Section 4.4 provides the details of our view-invariant dictionary learning algorithm. Section 4.5 presents the experimental results, and Section 4.6 concludes the chapter.

4.2 System Overview

As illustrated in Fig. 4.2 Left, in the pre-training phase, we synthesize 3D video sequences using motion capture data. We propose a new 3D dense trajectories feature extracted from a source 3D synthetic video, and $Y_{3D} = [\boldsymbol{y_{3D}}^1, ..., \boldsymbol{y_{3D}}^K] \in R^{S \times K}$ denotes the K S-dimensional features. The synthetic 3D video is projected into different viewpoints to create multiple synthetic 2D videos. $Y_{2D} = [\boldsymbol{y_{2D}}^1, ..., \boldsymbol{y_{2D}}^K] \in R^{T \times K}$ denotes the K T-dimensional features extracted from a target synthetic 2D video. We build 3D videos and 2D videos pairwisely in order to train the correspondence between them. We use K to denote both the numbers of 2D videos and 3D videos used in the pre-training phase.

We then train the 3D and 2D dictionaries simultaneously from the synthetic 3D and 2D videos respectively, which projects the respective video data into a common view-invariant sparse feature space. They are represented as $D_{3D} = [\boldsymbol{d_{3D}}^1, ..., \boldsymbol{d_{3D}}^N] \in R^{S \times N}$ and $D_{2D} = [\boldsymbol{d_{2D}}^1, ..., \boldsymbol{d_{2D}}^N] \in R^{T \times N}$, where N is the dimension of the sparse feature space. Records belonging to the same action class in both 3D and 2D data are constrained to share the same sparse representation. We construct the action classifier W in an end-to-end manner for better accuracy, by jointly minimizing the classification error rate and the dictionary quantization error. This improves training efficiency and system accuracy.

Then, as illustrated in Fig. 4.2 Middle, in the training phase, we replace the synthetic 2D videos with the 2D real training videos and perform system fine-tuning. This allows us to adapt the dictionaries ($D_{3D}{}'$, $D_{2D}{}'$) and the classifier (W') originally trained from synthetic data into real-world data. Because of the pre-training phase, only a small amount of real training videos are needed. We finally denormalize the 2D dictionary and the classifier into $\widehat{D}_{2D}{}'$ and $\widehat{W}'$ respectively.

In the testing phase illustrated in Fig. 4.2 Right, given any real 2D video, we apply $\widehat{D}_{2D}{}'$ to encode the features into a view-invariant sparse representation $X = [\boldsymbol{x}^1, ..., \boldsymbol{x}^K] \in R^{N \times K}$. We then apply $\widehat{W}'$ to identify the class label of the video. Due to the use of the view transfer dictionary, our system can identify actions from an arbitrary 2D view.

Fig. 4.3 shows the Overview of our baseline view-invariant human action recognition system in [3]. The major weakness of this framework is this is not an end-to-end system. A separate SVM classifier is required to recognise the actions.

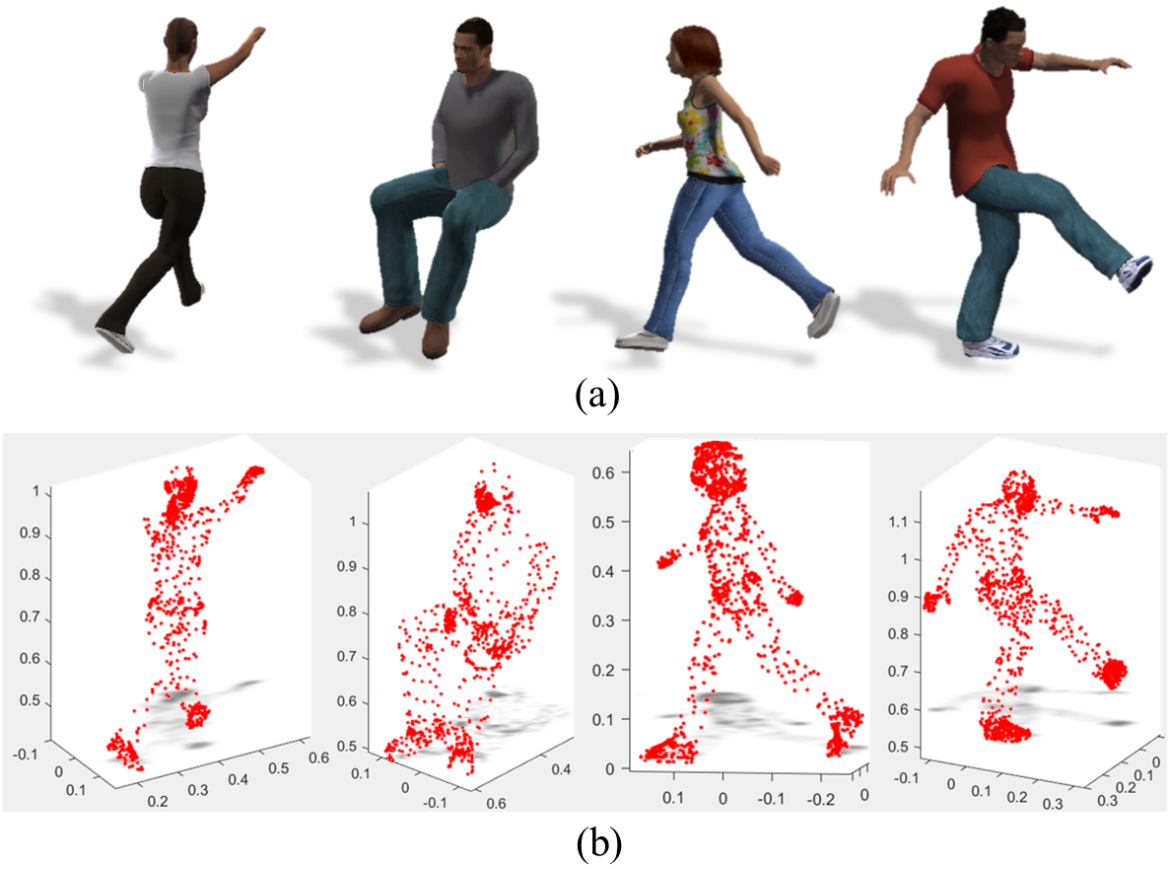


Fig. 4.4 (a) Some example frames from the synthetic 3D video. Using motion retargeting techniques, we can retarget the captured motion to 3D models of different body sizes to increase the database diversity. (b) The interest points obtained according to the vertices of the 3D models.

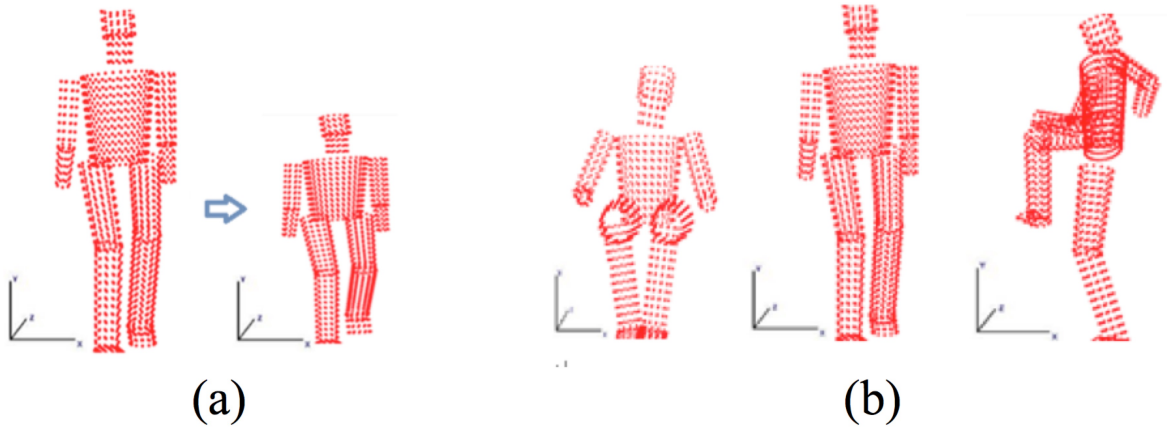


Fig. 4.5 (a) Some example frames from the baseline synthetic 3D video. We use cylinders to model body parts and represent surface information. (b) The interest points obtained according to the vertices of the 3D models.

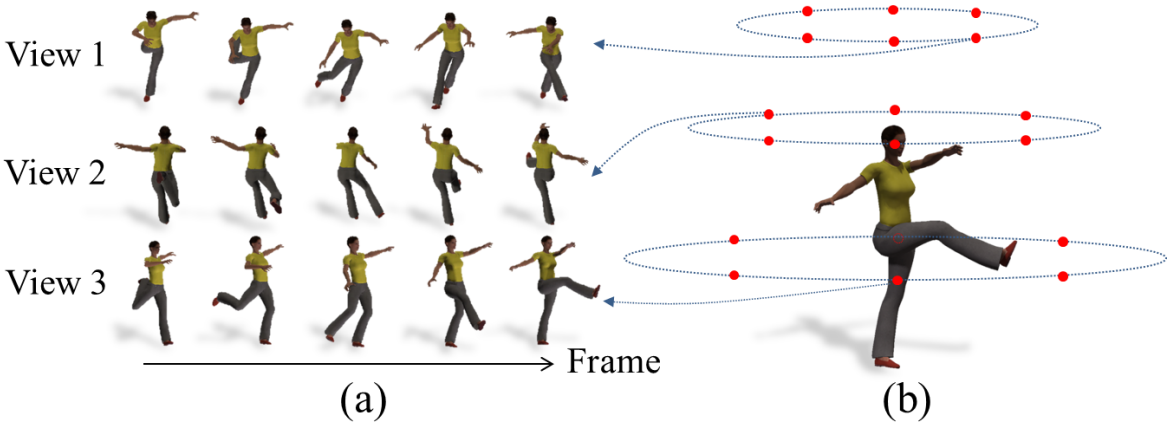


Fig. 4.6 (a) Example frames of synthetic 2D videos obtained by projecting a 3D video into different viewpoints. (b) Virtual cameras are placed on the hemisphere looking towards the center of the sphere to generate different viewpoints.

4.3 Video Synthesis and Feature Extraction

In this section, we explain how we synthesize 3D videos and project them to generate synthetic 2D videos. We then explain how we extract a corresponding set of 3D and 2D features.

4.3.1 Synthesizing 3D and 2D Videos

Here, we explain the process of synthesizing 3D and 2D video data.

To synthesize the 3D motion models, we utilize the motion capture data from the Carnegie-Mellon Graphics Lab and the Truebones dataset [129]. The motions are represented with 3D joint angles in a skeletal body hierarchy at 25 frames per second (FPS). Instead of using simplified cylindrical model to represent surface information as in past research [1, 3] (showed in Fig. 4.5), we use different high-resolution 3D human models instead. This requires a process known as *primary deformation* [127] to deform the human models based on the skeletal movement over time. The advantage of using 3D motion data is that we can apply *motion retargeting* techniques to synthesize the motion performed by human models of different body sizes, as shown in Fig. 4.4a. Such an automatic process enhances the diversity of the database by adjusting the movement according to the bone length.

In order to produce synthetic 2D video, we project the synthesized 3D videos uniformly in a set of pre-defined viewpoints. Fig. 4.6 shows example frames of 2D videos projected from various viewpoints. Notice that in our system, we do not require any information about the viewpoints to perform classification.

(a) (b)

Fig. 4.7 (a) Synthesized 2D video (b) Extracted dense trajectories (red points are interest points, green curves are trajectories)

4.3.2 2D Dense Trajectories

For both 2D synthetic videos and 2D real videos, we employ dense trajectories [42], a powerful action representation, for feature extraction. It considers both holistic and local information of 2D motion by combining dense sampling and trajectory tracking. Specifically, it consists of a set of low-level descriptors, including trajectory descriptor, Histogram of Oriented Gradients (HOG), Histogram of Optical Flow (HOF) and Motion Boundary Histogram (MBH). Among them, HOG can extract the static appearance of the videos while HOF and MBH can extract the motion information. Fig. 4.7 shows an example of dense trajectories extracted from a synthetic 2D video.

4.3.3 Proposed 3D Dense Trajectories

Our transfer learning involves transferring 3D and 2D features into a common sparse feature space, and hence it is preferable that both of them have similar logical meanings. Therefore, we propose a 3D version of dense trajectories that corresponds to the 2D one. The proposed feature consists of three components: 3D trajectories, 3DHOF and 3DMBH. Notice that HOG is not included here, as the surface texture of a 3D model remains unchanged over time.

An advantage of synthetic 3D videos is that both the vertices geometry on the human model surface and the vertices correspondence across frames are available. We first obtain a set of interest points over time according to the surface vertices of the 3D models, as shown

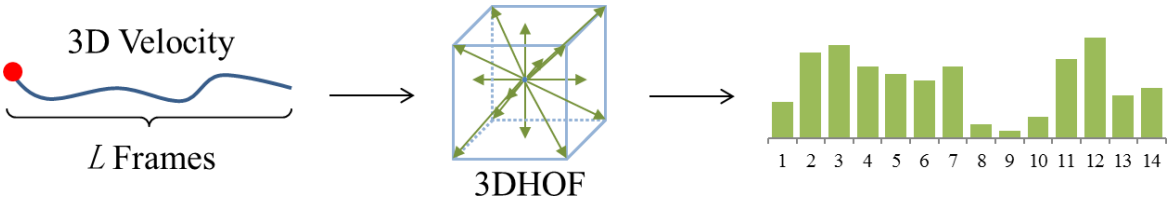


Fig. 4.8 The 14 3D velocity bins visualized with a 3D cube. 6 directions point towards the faces of the cube, and 8 directions point towards the corners of the cube.

in Fig. 4.4b. For each point, we extract the motion trajectory across frames $(P_t, P_{t+1}, P_{t+2}, ...)$, where P_t is the 3D Cartesian coordinate of the vertex at frame t, as shown in Fig. 4.1b.

The 3D trajectory is defined as:

$$Tr' = \frac{(\Delta P_t, ..., \Delta P_{t+L-1})}{\sum_{j=t}^{t+L-1} \|\Delta P_j\|} \tag{4.1}$$

where L is a user-defined value that represents the number of frames to be considered in a trajectory, and $\Delta P_t = (P_{t+1} - P_t)$ indicates the displacement across two frames. The denominator is the total length of the trajectory, which is used for normalization.

2D HOF is the pattern of apparent motion of objects and surface in a visual scene caused by the relative motion between an observer and a scene. A logically similar representation in 3D, which we named the 3D Histogram of Optical Flow (3DHOF), is the velocity field of the surface vertices. We first define the velocity of a vertex as:

$$V_t = \frac{\Delta P_t}{1/FPS} \tag{4.2}$$

where FPS is the frame rate of the 3D video, and is set to 25 in our experiments. We then quantize the 3D velocity orientations into 14 bins $H(h_1, h_2, ..., h_{14})$ as shown in Fig. 4.8. 3DHOF is defined as the binned histogram along each vertex trajectory:

$$h_i = \frac{\sum_{t \in T_i} \|V_t\|}{\sum_{j=t}^{t+L-1} \|V_t\|} \tag{4.3}$$

where T_i is a set that contains the frame's number in which the velocity direction of the interest point belongs to i on a L-frame trajectory. $\|V_t\|$ is the magnitude of the velocity, which is used for weighting.

The 2D MBH (motion boundary histogram) is the derivative of the optical flow field computed separately for the horizontal and vertical components to encode the relative motion between pixels. This is to compensate the HOF descriptor, which can only compute

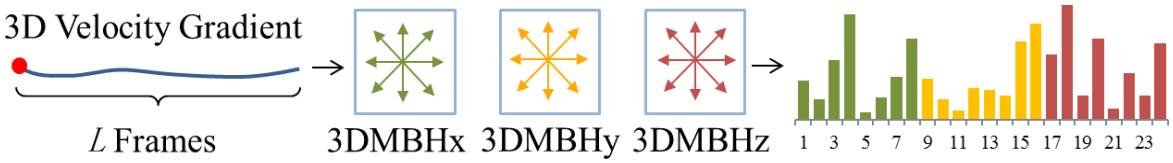


Fig. 4.9 The 3DMBH components in X, Y and Z directions are quantized into 8 bins each. The 3DMBH is defined as the concatenation of 3DMBHx, 3DMBHy and 3DMBHz along each vertex trajectory.

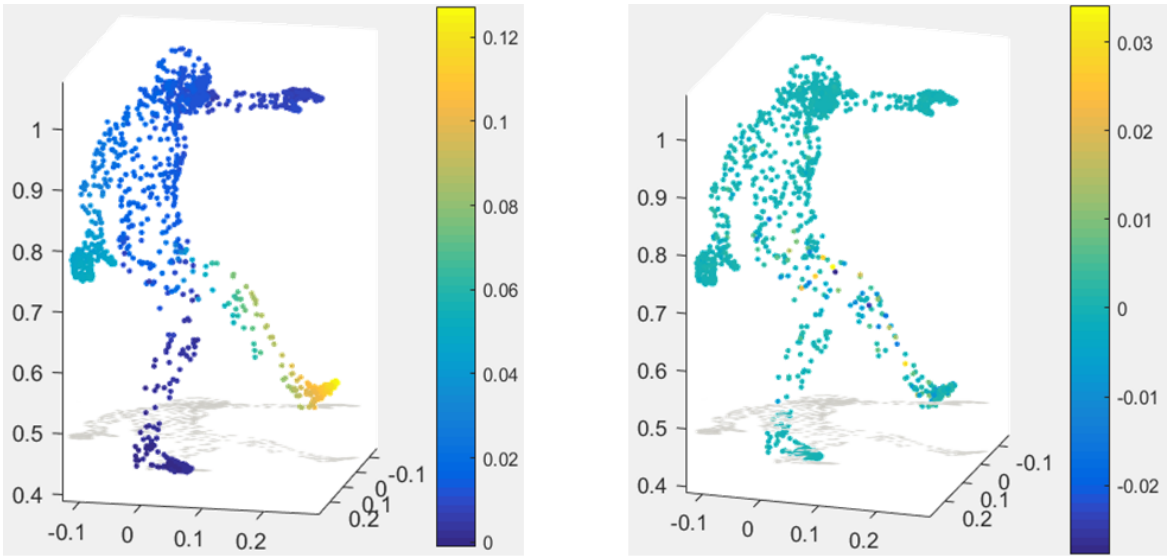


Fig. 4.10 (a) The Y component of 3D velocity field for the example frame. (b) 3DMBHy is obtained by computing the gradient of Y component of 3D velocity field.

absolute motion information. Inspired by this, we proposed the 3DMBH that encodes the relative motion between neighbour interest points on our 3D model. Similar to the 2D MBH implementation, we compute the derivatives separately along the X, Y, Z axes in the 3D velocity field. We quantize each 3DMBH component into 8 bins and the 3DMBH is defined as the concatenation of 3DMBHx, 3DMBHy and 3DMBHz along each vertex trajectory. The process is visualized as in Fig. 4.9. For example, the Y component of 3D velocity field is shown in Fig. 4.10a, and we compute its gradient to describe the relative motion between neighbouring interest points of that frame as shown in Fig. 4.10b.

4.4 View-invariant Action Classification

In this section, we explain how we train the view-invariant dictionaries and the classifier from synthetic 3D and 2D video data using dictionary learning. The processes are summarized as the algorithm shown in Fig. 4.11.

4.4.1 The Pre-training Phase

Here, we introduce the basic theory of dictionary learning [130], and explain how we learn the view-invariance transfer dictionary for the 3D and 2D synthetic videos.

Dictionary learning generates a sparse representation for a high dimensional signal using linear projection with a projection dictionary. Let $\boldsymbol{y} \in R^P$ denote a P-dimensional input signal that can be reconstructed by the Q-dimensional projection coefficient $\boldsymbol{x} \in R^Q$ via a linear projection dictionary $D = [\boldsymbol{d}^1, ..., \boldsymbol{d}^Q] \in R^{P \times Q}$. To obtain an over-completed dictionary, P should be much larger than Q. Assuming the reconstruction error to be $E(\boldsymbol{x})$, the projection process is formulated as:

$$\boldsymbol{y} = D\boldsymbol{x} + E(\boldsymbol{x}) \tag{4.4}$$

The objective function is defined as:

$$argmin_{\boldsymbol{x},D} \|\boldsymbol{y} - D\boldsymbol{x}\|_2^2 \qquad s.t. \|\boldsymbol{x}\|_0 \leq M \tag{4.5}$$

where $\|\boldsymbol{y} - D\boldsymbol{x}\|_2^2$ denotes the reconstruction error. $s.t. \|\boldsymbol{x}\|_0 \leq M$ denotes the sparsity constraint. M is the L_0-norm sparsity constraint factor that limits the number of non-zero elements in the sparse codes.

Due to the different number of trajectories across action videos, we use a bag-of-words descriptor to ensure that the features extracted from the action videos share the same dimension, following [40–43]. Specifically, we use K-means to cluster the trajectory-based

Input: 3D feature matrix Y_{3D}, 2D feature matrix Y_{2D}, target domain class label H, discriminative sparse code Q, sparsity constraint M, dictionary size N, trade-off parameter α, β, γ, iteration steps I.

Output: Dictionary $\widehat{D}_{3D}{}'$, $\widehat{D}_{2D}{}'$, linear classifier parameter $\widehat{W'}$.

Pre-training: (Synthetic 3D Video, Synthetic 2D Video)

Initialize D_{3D}, D_{2D}, A and W using K-SVD and Eq. 11, 12;

Reformulate $Y_0 = \begin{pmatrix} \sqrt{\alpha}Y_{3D} \\ Y_{2D} \\ \sqrt{\beta}Q \\ \sqrt{\gamma}H \end{pmatrix}$, $D_0 = \begin{pmatrix} \sqrt{\alpha}D_{3D} \\ D_{2D} \\ \sqrt{\beta}A \\ \sqrt{\gamma}W \end{pmatrix}$;

For $i = 1$ **to** I

// Sparse coding using OMP

$argmin_{X,D_0} \|Y_0 - D_0 X\|_2^2 \; s.t. \forall i, [\|\boldsymbol{x}^i\|_0] \leq M;$

// Dictionary updating using SVD

For $k = 1$ **to** N

$$argmin_{\boldsymbol{d_k},\tilde{\boldsymbol{x}}_k} \left\|\tilde{E}_k - \boldsymbol{d_k}\tilde{\boldsymbol{x}}_k\right\|_2^2$$

$$SVD(\tilde{E}_k) = U \sum V^T$$

$$\boldsymbol{d_k} = U(:,1)$$
$$\tilde{\boldsymbol{x}}_k = \sum(1,1)V(1,:);$$

End

Update D_0

End

Training: (Synthetic 3D Video, Real 2D Video)

Initialize $D_{3D}{}'$, $D_{2D}{}'$, A' and W' with pre-trained D_{3D}, D_{2D}, A and W;

Apply the same optimization strategy as the pre-training phase;

Denormalize the trained $D_{2D}{}'$ and W' to obtain $\widehat{D}_{2D}{}'$ and $\widehat{W}'$;

Testing: (Real 2D Video)

Use Eq. 14 for classification.

Fig. 4.11 The algorithm for transferring view-invariance from 3D video to 2D video by transfer dictionary learning.

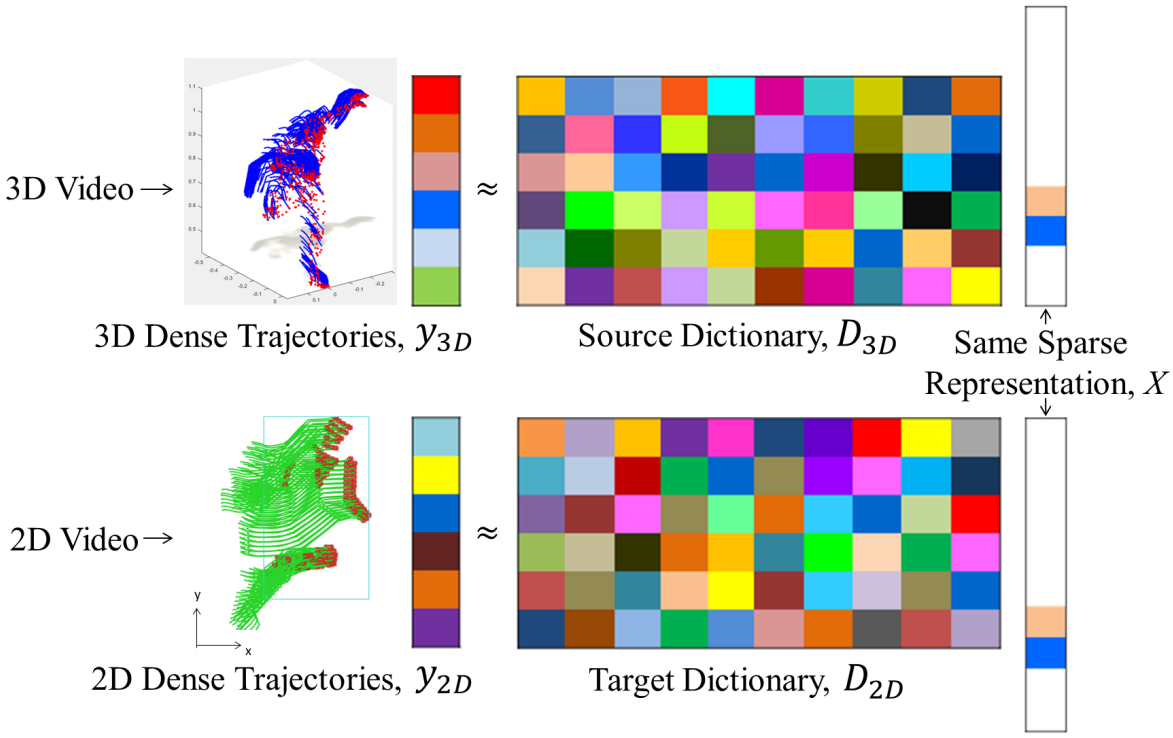


Fig. 4.12 Optimizing the 3D (source) and 2D (target) dictionaries to constraint that the same action in synthetic 3D and 2D videos has the same sparse representations.

descriptors in each action video into a fixed number of visual words. This allows us to represent the action videos with histograms of the same dimension.

We design a transfer dictionary learning system to transfer the view-invariance of the synthetic 3D videos to the synthetic 2D videos. We train two dictionaries simultaneously, with one for 3D (i.e. source - D_{3D}) and one for 2D (i.e. target - D_{2D}). The main idea is to optimize the dictionaries such that the same action in both 3D and 2D videos has the same sparse representations, as visualized in Fig. 4.12. Upon successful training, D_{2D} is able to project the feature vector of a 2D video into a sparse representation that is similar to that of a 3D video. In other words, such a sparse representation is view-invariant.

We divide the dictionary into a number of disjoint subsets, and each of these is used exclusively for one action category. 3D and 2D videos with the same action category are therefore represented by the same subset of the dictionary. Those with different action categories are represented with disjoint subsets of the dictionary. This design enables the 3D and 2D videos with the same action category to share the same sparse representation pattern. Conversely, those with different action categories tend to have different representations.

Specifically, the dictionary optimization function is designed as:

$$\begin{aligned} &argmin_{X,D_{3D},D_{2D},A} \\ &\alpha\|Y_{3D}-D_{3D}X\|_2^2+\|Y_{2D}-D_{2D}X\|_2^2+\beta\|Q-AX\|_2^2 \\ &s.t\forall i,\|\boldsymbol{x}^i\|_0\leq M \end{aligned} \tag{4.6}$$

where α and β are trade-off parameters, $\|Y_{3D}-D_{3D}X\|_2^2$ and $\|Y_{2D}-D_{2D}X\|_2^2$ are two terms to minimize the error of the 3D and 2D dictionaries respectively, and $\|Q-AX\|_2^2$ is a label consistent regularization term to minimize the difference in sparse representation for the same class of action as introduced in [131, 132]. A is a linear transformation matrix that maps the original sparse codes X to be consistent with the discriminative sparse codes $Q=[\boldsymbol{q}^1,...,\boldsymbol{q}^K]\in R^{N\times K}$ of input signal $(\boldsymbol{y_{3D}}^j,\boldsymbol{y_{2D}}^j)$, in which the index j indicates the index of 2D and 3D action video pairs. Specifically, each vector $\boldsymbol{q_j}=[\boldsymbol{q_j}^1,...,\boldsymbol{q_j}^N]=[0...1,1...0]\in R^N$, and the non-zero occurs at those indices where the input signal $(\boldsymbol{y_{3D}}^j,\boldsymbol{y_{2D}}^j)$ and the dictionary items $(\boldsymbol{d_{3D}}^n,\boldsymbol{d_{2D}}^n)$ share the same label. In our dictionary design, dictionary item $\boldsymbol{d_{3D}}^n$ and $\boldsymbol{d_{3D}}^n$ always have the same label. For example, assuming the $Y_{2D}=[\boldsymbol{y_{2D}}^1,...,\boldsymbol{y_{2D}}^6]$ and $D_{2D}=[\boldsymbol{d_{2D}}^1,...,\boldsymbol{d_{2D}}^6]$, where $\boldsymbol{y_{2D}}^1,\boldsymbol{y_{2D}}^2$ and $\boldsymbol{d_{2D}}^1,\boldsymbol{d_{2D}}^2$ are from class 1, $\boldsymbol{y_{2D}}^3,\boldsymbol{y_{2D}}^4$ and $\boldsymbol{d_{2D}}^3,\boldsymbol{d_{2D}}^4$ are from class 2, $\boldsymbol{y_{2D}}^5,\boldsymbol{y_{2D}}^6$ and $\boldsymbol{d_{2D}}^5,\boldsymbol{d_{2D}}^6$ are from class 3, then Q can be defined as:

$$Q=\begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \tag{4.7}$$

Inspired by [132], we propose to include the action classification error of a linear prediction classifier into the object function to build an end-to-end system. This enhances the system training efficiency and results in better classification accuracy. The new objective function is therefore updated as

$$\begin{aligned} &argmin_{X,D_{3D},D_{2D},A,W} \\ &\alpha\|Y_{3D}-D_{3D}X\|_2^2+\|Y_{2D}-D_{2D}X\|_2^2+\beta\|Q-AX\|_2^2 \\ &+\gamma\|H-WX\|_2^2 \qquad s.t.\forall i,\|\boldsymbol{x}^i\|_0\leq M \end{aligned} \tag{4.8}$$

where $\|H - WX\|_2^2$ is the proposed action classification error term, $W \in R^{C \times N}$ denotes the classifier parameters and $H = [\boldsymbol{h}^1, ..., \boldsymbol{h}^K] \in R^{C \times K}$ are the class label of input signals Y_{2D}. $\boldsymbol{h}^j = [0...1...0]^T \in R^C$ is a label vector corresponding to an input signal $\boldsymbol{y_{2D}}^j$, where the nonzero position indicates the class of $\boldsymbol{y_{2D}}^j$.

4.4.2 Optimization

Here, we explain how we obtain the solution for Eq. 4.8. Since the three terms on the right hand side of Eq. 4.8 have the same format, we first rewrite Eq. 4.8 as follows:

$$argmin_{X,D_0} \|Y_0 - D_0 X\|_2^2 \qquad s.t. \forall i, \|\boldsymbol{x}^i\|_0 \leq M \tag{4.9}$$

where $Y_0 = \begin{pmatrix} \sqrt{\alpha} Y_{3D} \\ Y_{2D} \\ \sqrt{\beta} Q \\ \sqrt{\gamma} H \end{pmatrix}$, $D_0 = \begin{pmatrix} \sqrt{\alpha} D_{3D} \\ D_{2D} \\ \sqrt{\beta} A \\ \sqrt{\gamma} W \end{pmatrix}$.

Such an objective function shares the same form as Eq. 4.5, which can be optimized using the K-SVD algorithm [79]. Specifically, Eq. 4.9 is solved through both dictionary atom updating and sparse representing.

For the dictionary atom updating stage, each dictionary atom is updated sequentially to better represent both 3D videos and 2D videos. When pursing the better dictionary D_0, the sparse representation X is fixed, and each dictionary atom is updated by tracking down a rank-one approximation to the matrix of residuals.

Following K-SVD, the k^{th} atom of dictionary D_0 and its corresponding coefficients are denoted as $\boldsymbol{d_k}$ and $\boldsymbol{x_k}$ respectively. Let $E_k = Y_0 - \sum_{j \neq k} \boldsymbol{d_j x_j}$ and we further denote $\widetilde{\boldsymbol{x}}_{\boldsymbol{k}}$ and $\widetilde{E}$ as the result obtained when all zero entries in $\boldsymbol{x_k}$ and E_k are discarded respectively. Each dictionary atom $\boldsymbol{d_k}$ and its corresponding non-zero coefficients $\widetilde{\boldsymbol{x}}_{\boldsymbol{k}}$ can be computed by:

$$argmin_{\boldsymbol{d_k}, \widetilde{\boldsymbol{x}}_{\boldsymbol{k}}} \|\widetilde{E}_k - \boldsymbol{d_k} \widetilde{\boldsymbol{x}}_{\boldsymbol{k}}\|_2^2 \tag{4.10}$$

The approximation in Eq. 4.10 is achieved through Singular Value Decomposition (SVD) on $\widetilde{E}_k$:

$$\begin{aligned} SVD(\widetilde{E}_k) &= U \sum V^T \\ \boldsymbol{d_k} &= U(:,1) \\ \widetilde{\boldsymbol{x}}_{\boldsymbol{k}} &= \sum(1,1) V(1,:) \end{aligned} \tag{4.11}$$

where $U(:,1)$ indicates the first column of U while $V(1,:)$ indicates the first row of V.

At the sparse representation stage, we compute the best matching projection X of the multidimensional training data for the updated dictionary D_0 using Orthogonal Matching Pursuit (OMP) algorithm.

Initialization

D_{3D}, D_{2D}, A and W must be initialized before pre-training. In our system, for D_{3D} and D_{2D}, we run a few iterations of K-SVD within each action class and initialize the label of the dictionary items based on the corresponding action labels. To initialize A and W, we use the multivariate ridge regression model [133] with the L_2-norm:

$$\begin{aligned} A &= argmin_A \|Q - AX\|_2^2 + \varphi_1 \|A\|_2^2 \\ W &= argmin_W \|H - WX\|_2^2 + \varphi_2 \|W\|_2^2 \end{aligned} \tag{4.12}$$

where φ_1 and φ_2 are manually defined constants and are empirically set as 0.5 in our system. The equation yields the following solutions:

$$\begin{aligned} A &= (XX^t + \varphi_1 I)^{-1} \\ W &= (XX^t + \varphi_2 I)^{-1} \end{aligned} \tag{4.13}$$

where X is calculated with the initialized D_{3D} or D_{2D}.

Convergence Analysis

The convergence proof of the proposed method is similar with the K-SVD algorithm. In the dictionary updating stage, each atom d_k and its corresponding coefficients $\tilde{x}_k$ minimize the objective function, while the rest of dictionary atoms are updated iteration by iteration. Therefore, the Mean Squared Error (MSE) of the reconstruction error should be monotonically decreasing. At the sparse representation stage, the MSE is also reduced due to the computation of the best matched coefficients under the L_0-norm constraint of the OMP algorithm. In addition, since MSE is non-negative, the optimization process should be monotonically reducing and bounded by zero. Therefore, the convergence of the proposed transfer dictionary learning method is guaranteed. We also show the convergence of the system in Fig. 4.15.

4.4.3 The Training Phase

Here, we explain how to adapt the pre-trained dictionaries and classifier into real video.

We fine-tune the dictionaries and the classifier pre-trained by the synthetic data in order to adapt them into real-world data. Specifically, we use D_{3D}, D_{2D}, A, W in the pre-training phase to initialize the training phase. We also replace the 2D synthetic videos with 2D real training videos. Then, we follow the same optimization strategy in Section 4.4.2 and apply

the same number of iterations as the pre-training phase. After the optimization, we denote the trained dictionaries and classifiers as ($D_{3D}{}'$, $D_{2D}{}'$) and W' respectively.

Since $D_{3D}{}'$, $D_{2D}{}'$, W' are jointly $L2$-normalized during the optimization process, we need a step of de-normalization before they can be used for classification. Following [132], the denormalized 2D dictionary $\widehat{D}_{2D}{}'$ and the classification parameter $\widehat{W}'$ are calculated as:

$$\begin{aligned} \widehat{D}_{2D}{}' &= \left(\frac{\boldsymbol{d_{2D\_1}}'}{\|\boldsymbol{d_{2D\_1}}'\|_2}, \frac{\boldsymbol{d_{2D\_2}}'}{\|\boldsymbol{d_{2D\_2}}'\|_2}, ..., \frac{\boldsymbol{d_{2D\_N}}'}{\|\boldsymbol{d_{2D\_N}}'\|_2} \right) \\ \widehat{W}' &= \left(\frac{\boldsymbol{w_1}'}{\|\boldsymbol{w_1}'\|_2}, \frac{\boldsymbol{w_2}'}{\|\boldsymbol{w_2}'\|_2}, ..., \frac{\boldsymbol{w_N}'}{\|\boldsymbol{w_N}'\|_2} \right) \end{aligned} \tag{4.14}$$

where $\boldsymbol{d_{2D\_n}}'$ denotes the n^{th} atom of the dictionary $D_{2D}{}'$, $\boldsymbol{w_N}'$ denotes the n^{th} atom of W'. Notice that we do not denormalize $D_{3D}{}'$ as it is no longer needed in the next phase.

4.4.4 The Testing Phase

Here, we explain how we apply our trained dictionary to perform view-invariant action classification.

Given a real 2D video query sample $\boldsymbol{y_{2D}}'$, its sparse representation $\boldsymbol{x}'$ can be computed with $\widehat{D}_{2D}{}'$. With the linear classification parameter $\widehat{W}'$, the label $\boldsymbol{l}$ can be predicted as:

$$\boldsymbol{l} = \widehat{W}'\boldsymbol{x}' \tag{4.15}$$

The label of $\boldsymbol{y_{2D}}'$ is the index corresponding to the largest element of $\boldsymbol{l}$.

4.5 Experimental Results

In this section, we first provide experiment setup details. We then evaluate the performance of our method with four public multi-view datasets including the IXMAS, N-UCLA, UWA3DII and i3DPost datasets.

The synthetic 3D and 2D datasets we used for transfer dictionary learning are open to the public. They can be found at our project website. All experiments were performed on a desktop computer with an Intel i7-4790k CPU, a NVIDIA Quadro K2200 graphics card, 16GB RAM and Microsoft Windows 10 OS.

4.5.1 Implementation Details

We used the software package Poser 2014 to retarget 3D motion capture data files in BVH format, animate 3D human models, and project the 3D scenes into 2D videos. We employed 5 high-quality 3D characters to synthesize the 3D video. For each action class, we synthesized 18 3D videos per character with 18 randomly selected motion files within the class. For each action class, we synthesized the 2D videos per character by projecting a randomly selected 3D video into 18 uniformly sampled viewpoints. The azimuthal angle of the projection was uniformly sampled as $\{0^\circ, 60^\circ, 120^\circ, 180^\circ, 240^\circ, 300^\circ\}$ and the polar angle of the projection was sampled as $\{0^\circ, -30^\circ, -60^\circ\}$. This setup allowed us to generate the same number of 3D and 2D videos (number of characters $\times$ number of views $\times$ number of action classes)as required by K-SVD for transfer dictionary learning.

During pre-training, for the experiments on the IXMAS dataset (11 action classes), the N-UCLA dataset (10 action classes), the i3DPost dataset (10 action classes) and the UWA3DII dataset (30 action classes), we synthesized 990, 900, 900, 2700 pairwise 3D and 2D videos, respectively. From our experience, a larger synthetic dataset resulted in better accuracy. The size used was chosen considering the trade-off between system accuracy and training complexity.

We extracted dense trajectories from 2D synthetic videos, as well as 2D real videos from the IXMAS, N-UCLA, UWA3DII and i3DPost datasets. Afterwards, we constructed a codebook for each of the four descriptors in the dense trajectories separately. For each 2D descriptor, we applied k-means to cluster a subset of 100,000 dense trajectory features into 375 visual words. This resulted in a 2D feature Y_{2D} of 1,500 dimensions. For 3D synthetic videos, similar to [53], we set the trajectory sample step to 5 frames, and the trajectory length to 15 frames. We constructed codebooks for 3D trajectories, 3DHOF and 3DMBH descriptors respectively. For each 3D descriptor, we applied k-means to cluster a subset of 100,000 3D dense trajectories into 500 visual words. This resulted in a 3D feature Y_{3D} of 1,500 dimensions.

When training the transfer dictionaries, to initialize the dictionary pair D_{3D} and D_{2D}, we employed k-means 5 times on the features Y_{3D} and Y_{2D} respectively. For IXMAS, N-UCLA, UWA3DII and i3DPost datasets, we set the dictionary sizes N to 1180, 1150, 2500 and 1150 respectively, for both D_{3D} and D_{2D}. The 3D dictionary trade-off parameter α was set to 1.5. The label consistent trade-off parameter β was set to be 2.0. The classification error trader-off parameter γ was set to be 4.0. Finally, the numbers of iterations for the K-SVD algorithm in both pre-training and training phases were set to 60, 65, 100 and 65 for IXMAS, N-UCLA, UWA3DII and i3DPost datasets respectively.

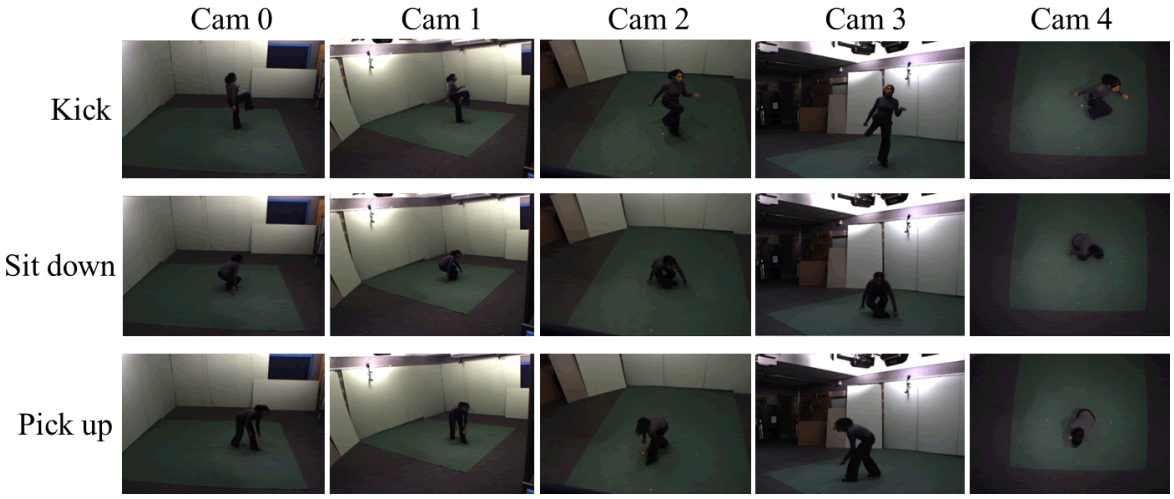


Fig. 4.13 Sampled frames from the IXMAS dataset.

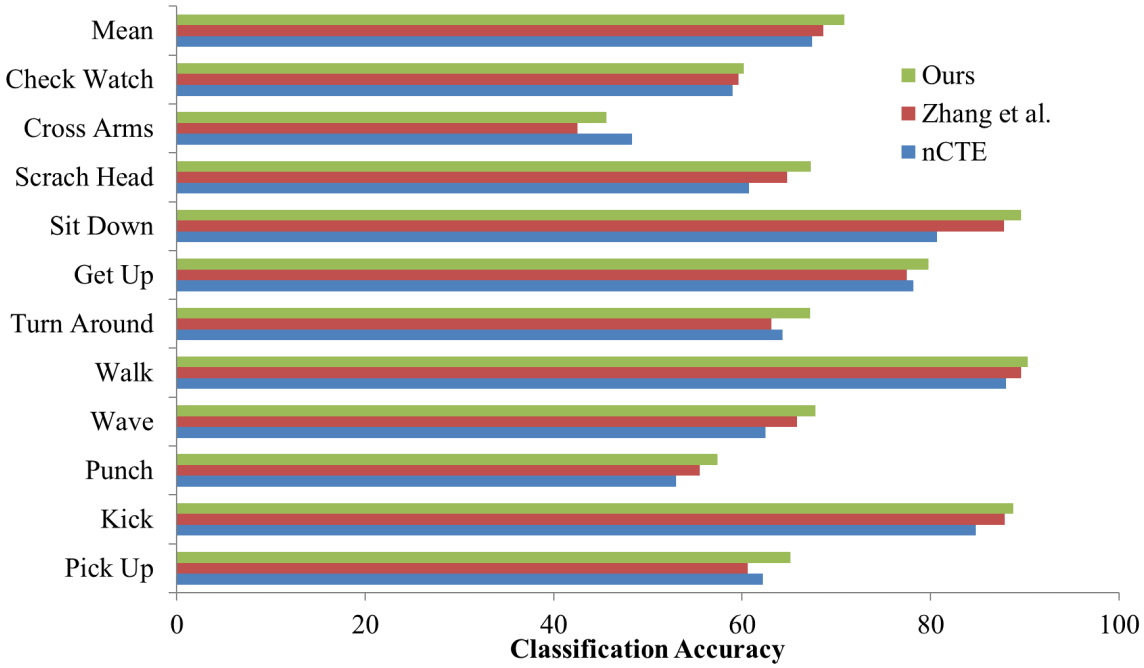


Fig. 4.14 Cross-view recognition accuracy per action class in IXMAS.

Methods	0|1	0|2	0|3	0|4	1|0	1|2	1|3	1|4	2|0	2|1	2|3	2|4	3|0	3|1	3|2	3|4	4|0	4|1	4|2	4|3	Mean
DVV[134]	72.4	13.3	53.0	28.8	64.9	27.9	53.6	21.8	36.4	40.6	41.8	37.3	58.2	58.5	24.2	22.4	30.6	24.9	27.9	24.6	56.4
CVP[135]	78.5	19.5	60.4	33.4	67.9	29.8	55.5	27.0	41.0	44.9	47.0	41.0	64.3	62.2	24.3	26.1	34.9	28.2	29.8	27.6	38.2
nCTE[1]	94.8	69.1	83.9	39.1	90.6	79.7	79.1	30.6	72.1	86.1	77.3	62.7	82.4	79.7	70.9	37.9	**48.8**	40.9	70.3	49.4	67.3
Hankelets[136]	83.7	59.2	57.4	33.6	84.3	61.6	62.8	26.9	62.5	65.2	72.0	60.1	57.1	61.5	71.0	31.2	39.6	32.8	68.1	37.4	56.4
Zhang et al.[3]	91.7	70.2	84.7	**44.4**	**92.3**	**81.4**	84.1	45.4	66.5	**87.3**	75.5	58.7	84.3	80.9	66.7	45.8	32.4	48.9	**74.8**	**53.3**	68.5
Without pre-training	86.1	68.8	74.7	34.6	81.4	74.6	78.4	37.9	68.4	78.6	73.5	58.3	76.5	72.3	64.3	48.7	31.7	37.1	67.8	41.1	62.7
Ours	**96.2**	**71.3**	**85.2**	41.5	90.6	80.7	**89.7**	**47.5**	**74.2**	85.3	**82.1**	**60.5**	**85.1**	**84.9**	**73.5**	**57.6**	41.6	**52.8**	71.6	50.8	**71.1**

Table 4.1 Cross-view recognition accuracy of all possible viewpoint combinations on IXMAS database. The horizontal axis labels are formatted as “source view|target view”.

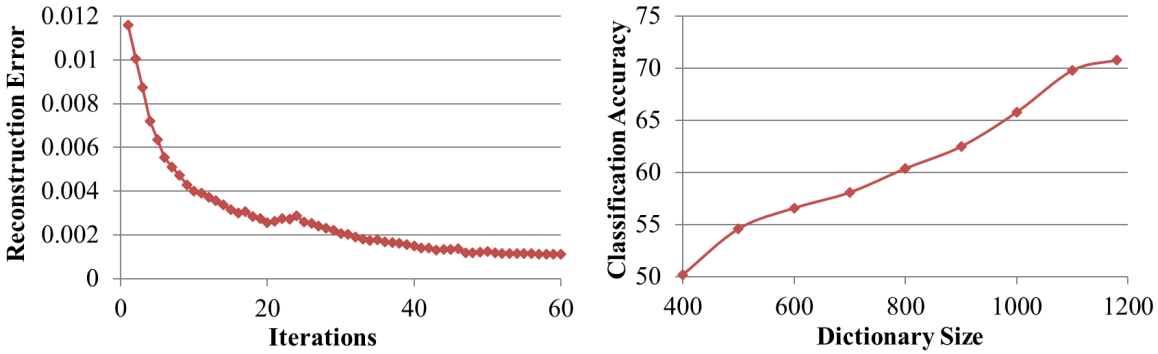


Fig. 4.15 Parameter analysis on the cross-view action recognition in IXMAS dataset. (a) The optimization process of the objective function with 50 iterations. (b) Performance with varying the dictionary size.

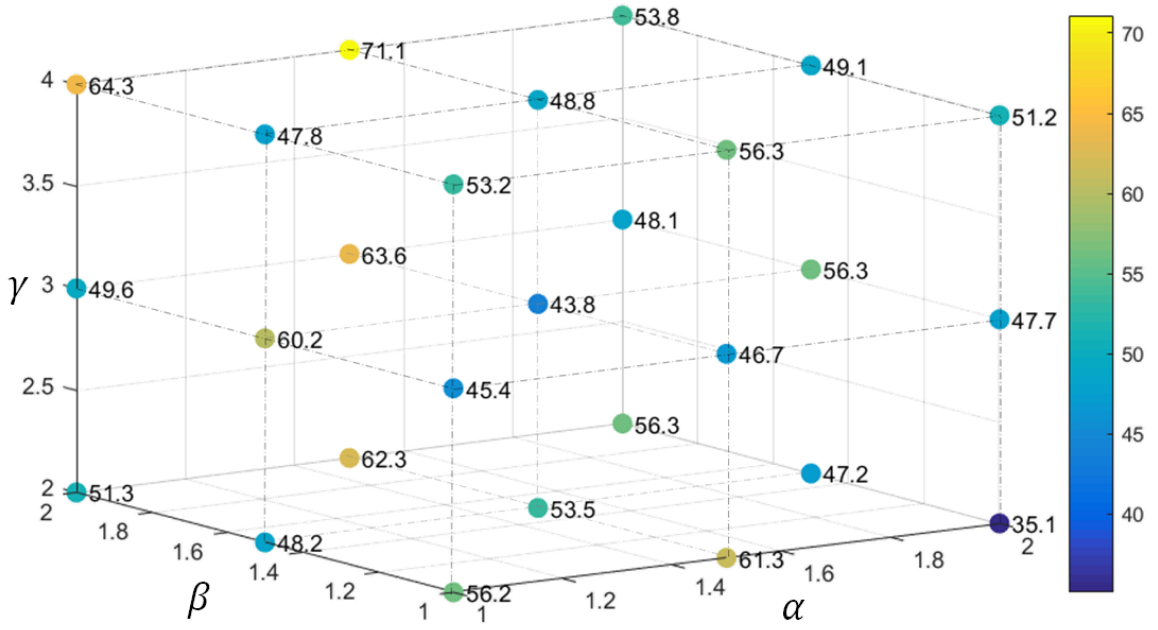


Fig. 4.16 Analysis on hyperparameters in Equation 7.

Methods	C0	C1	C2	C3	C4
DVV[134]	44.7	45.6	31.2	42.0	27.3
CVP[135]	50.0	49.3	34.7	45.9	31.0
nCTE[1]	72.6	72.7	73.5	70.1	47.5
Hankelets[136]	59.7	59.9	65.0	56.3	41.2
NKTM[2]	77.8	75.2	80.3	74.7	54.6
R-NKTM[54]	**78.4**	78.0	**80.7**	75.8	**57.8**
Zhang et al.[3]	70.8	76.5	72.6	71.9	50.5
Ours	73.2	**78.5**	74.9	**76.1**	52.9

Table 4.2 Average accuracy on the IXMAS dataset for each camera, e.g. C0 is the average accuracy when camera 0 is used for training or testing. Each time, only one camera view is used for training and testing.

4.5.2 Experiments on the IXMAS Dataset

The IXMAS dataset [59] contains 11 daily-life actions including “check watch”, “cross arms”, “scratch head”, “sit down”, “get up”, turn around”, “walk”, “wave”, “punch”, “kick”, and “pick up”. Each action was performed three times by 10 subjects captured from 5 different viewpoints. Fig. 4.13 shows some examples.

In order to compare with existing works on cross-view action recognition that utilize view labels including DVV [134], CVP [135], nCTE [1], Hankelets [136], and our preliminary work [3], we conducted an experiment considering view labels. Here, we grouped the videos in the IXMAS dataset into different views and evaluated the accuracy of transferring one view to another. We followed the leave-one-action-out cross-validation strategy from [1, 136]. Table 4.1 shows that our algorithm outperforms the state-of-the-art method nCTE in most cross-view pairs, as well as the average system accuracy. It also demonstrates that our proposed methodology enhancements over [3] have resulted in superior accuracy. We also compare with a baseline setup of our system that does not include the pre-training phase, which demonstrates the effectiveness of utilizing synthetic 2D and 3D videos for pre-training. Fig. 4.14 shows that our algorithm outperforms nCTE in most action classes, thereby indicating that our system can can realize cross-view action recognition by transferring the view-invariance from 3D models. Notice that in our default setup, the system does not require any view information. This experiment was designed for the sake of comparison only.

In order to analyze the effect of the hyperparameters (i.e. α, β and γ), we experiment with 27 different settings within the searching range of α in [1, 2] on every 0.5 interval, β in [1, 2] on every 0.5 interval and γ in [2, 4] on every 1.0 interval. The result is visualized in Fig. 4.16.

Fig. 4.17 Sampled frames from the N-UCLA dataset.

Since the orientation of the actors is arbitrary in the IXMAS dataset, we compare with existing works on arbitrary view action recognition by calculating average accuracy for each camera. For example, C0 is the average accuracy when camera0 is used for training or testing. Table 4.2 shows that our algorithm outperforms most of the previous methods in some viewpoints. It is worth mentioning that NKTM [2] and R-NKTM [54] are deep learning based methods, at the core of which is the use of neural networks to transfer videos from different views to a canonical view. However, their method requires the generation of 2D training video by projecting the 3D exemplar to 108 virtual views, while ours only needs 18 different views. Due to the lower amount of training data required, our method can save computation resources, especially when constructing the system.

4.5.3 Experiments on the N-UCLA Dataset

The N-UCLA dataset [51] contains 10 action classes captured from 3 different viewpoints with 10 different actors. The action categories include “pick up with one hand”, “pick up with two hands”, “drop trash”, “walk around”, “sit down”, “stand up”, “donning”, “doffing”, “throw”, and “carry”. Fig. 4.17 shows some sample frames from the N-UCLA dataset.

We evaluated our system accuracy in cross-view action recognition and in comparison with existing work including DVV [134], nCTE [1], CVP [135], and our preliminary work

Methods	{1,2}|3	{1,3}|2	{2,3}|1	Mean
DVV[134]	58.5	55.2	39.3	51.0
CVP[135]	60.6	55.8	39.5	52.0
nCTE[1]	68.6	68.3	52.1	63.0
Zhang et al.[3]	67.3	74.2	61.8	67.8
Ours	**69.1**	**74.4**	**61.8**	**68.5**

Table 4.3 Accuracy on the N-UCLA dataset (two views for training and one for testing).

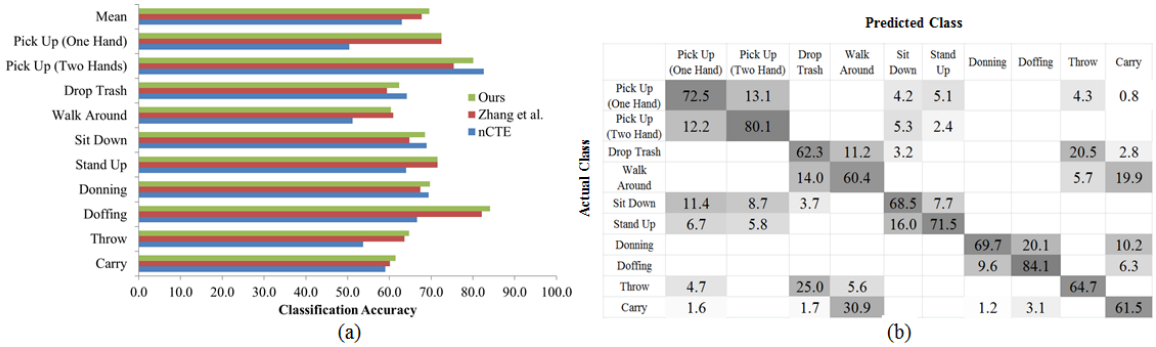


Fig. 4.18 (a) Cross-view recognition accuracy per action class in N-UCLA. (b) The confusion matrix of N-UCLA.

[3]. We followed the experimental setup in [1, 135], which utilizes videos captured from two cameras for training and the other one for testing. The accuracy was calculated using leave-one-action-out cross validation. As shown in Table 4.3, our method outperforms existing algorithms in most of the cross-view setups and the overall result. Fig. 4.18 shows that our algorithm outperforms nCTE in most action classes. This demonstrates that our system can realize cross-view action recognition by transferring the view-invariance from 3D models. Notice that in our default setup, the system does not require view information. This experiment was designed for the sake of comparison only.

On the N-UCLA dataset, some actions are quite difficult to differentiate, such as "Drop Trash" vs. "Throw", "Carry" vs. "Walk around", as they both consist of similar body movement.

4.5.4 Experiments on the UWA3DII Dataset

This dataset [137] consists of a variety of daily-life human actions performed by 10 subjects with different scales. It includes 30 action classes: "one hand waving", "one hand punching", "two hand waving", "two hand punching", "sitting down", "standing up", "vibrating", "falling down", "holding chest", "holding head", "holding back", "walking", "irregular walking",

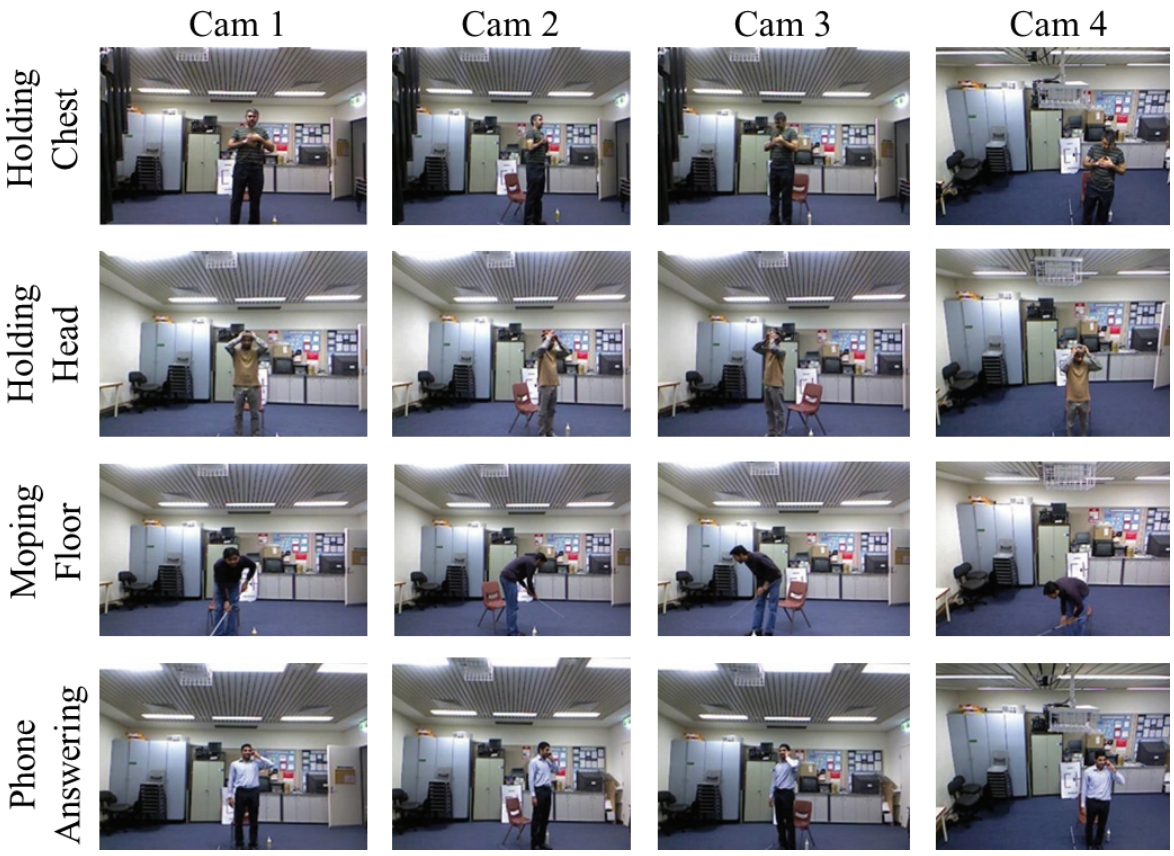


Fig. 4.19 Sampled frames from the UWA3DII dataset.

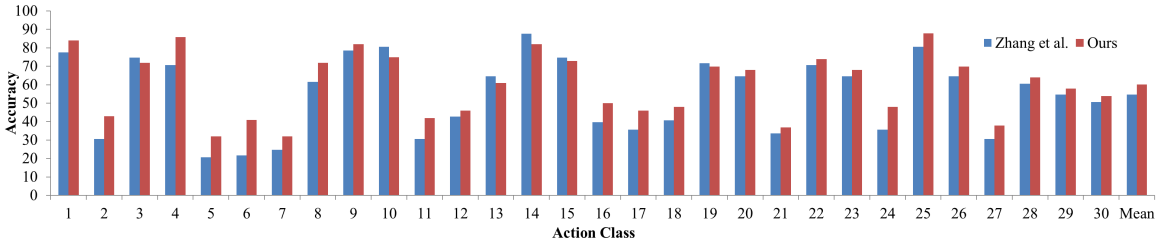


Fig. 4.20 Cross-view recognition accuracy per action class in the UWA3DII dataset.

"lying down", "turning around", "drinking", "phone answering", "bending", "jumping jack", "running", "picking up", "putting down", "kicking", "jumping", "dancing", "moping floor", "sneezing", "sitting down (chair)", "squatting", and "coughing". Each video is captured from one of four predefined viewpoints. This results in variations in actions across different viewpoints within the same action class. This dataset is challenging because of varying actor orientations, self-occlusion and high similarity among actions. Fig. 4.19 shows four sample actions from different viewpoints.

As shown in Table 4.4, our method outperforms existing algorithms in most of the cross-view setups and the overall result. Fig. 4.20 shows that our algorithm outperforms our baseline in most action classes.

4.5.5 Experiments on the i3DPost Dataset

The i3DPost dataset consists of 8 actors performing 10 different actions, where 6 are single actions: "walk", "run", "jump", "bend", "hand-wave" and "jump-in-place", and 4 are combined actions: "sit-stand-up", "run-fall", "walk-sit" and "run-jump-walk". The subjects have different body sizes, clothing and are of different sex and nationalities. The multi-view videos have been recorded by 8 calibrated and synchronized cameras in a high definition resolution (1920×1080), resulting in a total of 640 videos. For each video frames, an actor 3D mesh model of high detail level (20000-40000 vertices and 40000-80000 triangles) and the associated camera calibration parameters are available. The mesh models were reconstructed using a global optimization method proposed by Starck and Hilton [137]. Fig. 4.21 shows multi-view actor/action examples from the i3DPost dataset.

We use leave-one-actor out strategy followed by [140]. This means that we use the 2D videos of one actor for testing, while using the rest of the dataset for training. Table 4.5 shows that our system achieves better result than previous methods.

Methods	{1,2}|3	{1,2}|4	{1,3}|2	{1,3}|4	{1,4}|2	{1,4}|3	{2,3}|1	{2,3}|4	{2,4}|1	{2,4}|3	{3,4}|1	{3,4}|2	Mean
AOG[51]	47.3	39.7	43.0	30.5	35.0	42.2	50.7	28.6	51.0	43.2	51.6	44.2	42.3
Action Tube[138]	49.1	18.2	39.6	17.8	35.1	39.0	52.0	15.2	47.2	44.6	49.1	36.9	37.0
LRCN[139]	53.9	20.6	43.6	18.6	37.2	43.6	56.0	20.0	50.5	44.8	53.3	41.6	40.3
Zhang et al.[3]	50.6	56.8	48.6	43.7	53.2	59.7	**66.8**	48.9	56.8	50.4	68.3	51.7	54.6
Ours	**59.3**	**57.9**	**50.2**	**48.1**	**59.9**	**63.4**	65.1	**67.1**	**68.2**	**55.5**	**73.5**	**53.4**	**60.1**

Table 4.4 Accuracy on the UWA3DII dataset (two views for training and one for testing).

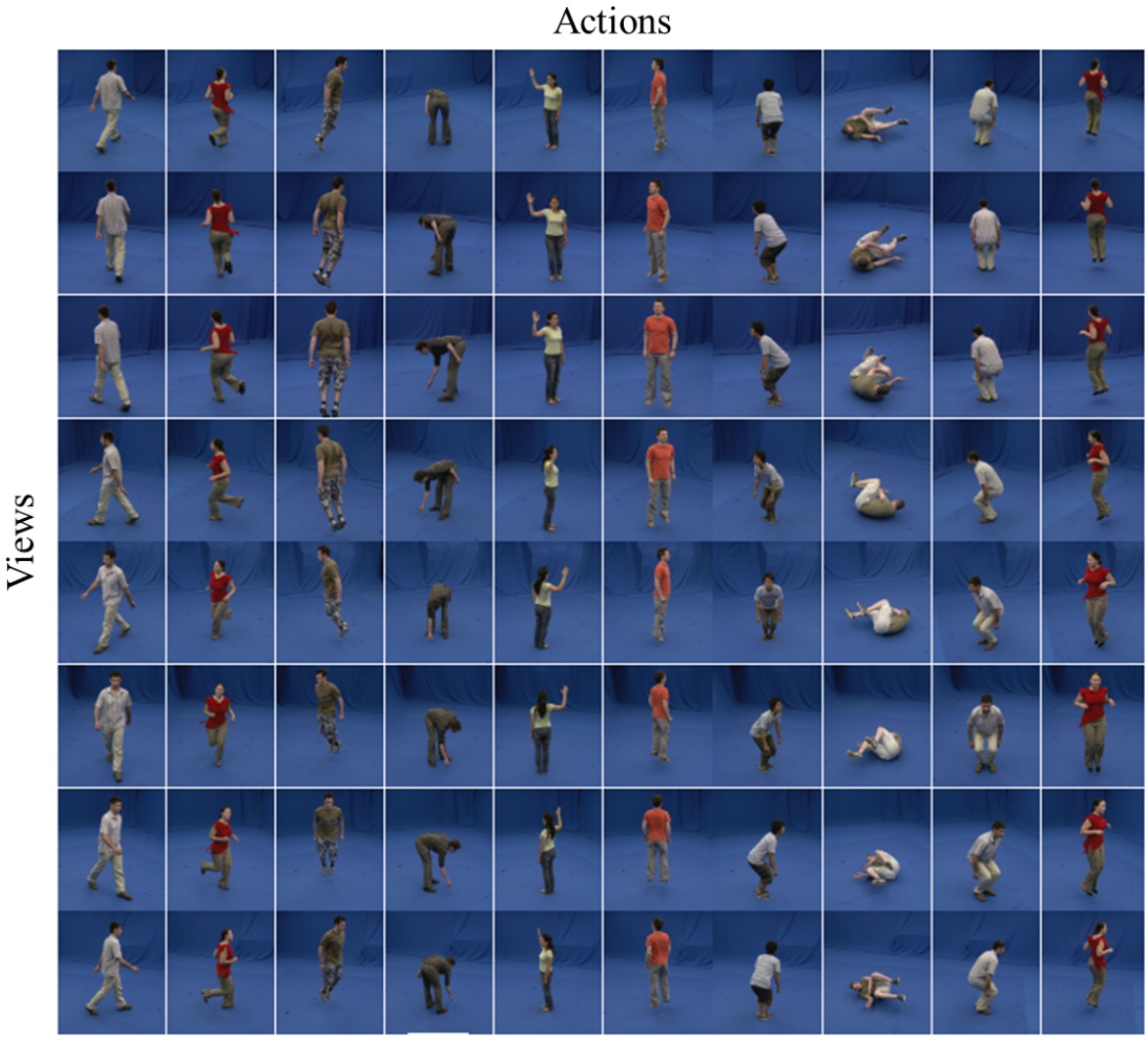


Fig. 4.21 Sampled frames from the i3DPost dataset.

Methods	Mean
Holte et al.[140]	92.2
Iosifidis et al.[141]	90.9
Gkalelis et al.[137]	90.0
Zhang et al.[3]	93.8
Ours	**94.6**

Table 4.5 Average accuracy for arbitrary view recognition on the i3DPost dataset.

Features	IXMAS	N-UCLA	UWA3DII	i3DPost
3D Trajectories	58.1	57.1	48.6	90.1
3DHOF	67.8	62.4	58.4	87.5
3DMBH	66.3	56.3	53.2	81.6
All Features Combined	**70.8**	**68.5**	**60.1**	**94.6**

Table 4.6 Comparison of cross-view action recognition results on the IXMAS, N-UCLA, UWA3DII and i3DPost dataset by using different features.

4.5.6 Evaluation of our 3D Dense Trajectories

In this section, we evaluate our 3D dense trajectories by using 3D trajectories, 3DHOF and 3DMBH independently.

Table 4.6 shows the comparison of cross-view action recognition results on the IXMAS, N-UCLA, UWA3DII and i3DPost dataset by using each descriptor independently and combining them together. Among the three descriptors, 3DHOF outperforms the other two in the most of dataset. However, it is clear that the combined feature produces far superior results that cannot be achieved by any single feature. This shows that our proposed features are complementary to each other.

Fig. 4.22 and 4.23 show the performance of different descriptors according to different view transfer pairs on the IXMAS and UWA3DII datasets respectively. In all pairs, combining all the descriptors achieves better result than using them independently.

4.5.7 Evaluation of 2D Features Used in Our System

While appearance information and movement information are both very important for describing the 2D action videos, such appearance information is quite different for 2D action videos captured from different points. We build a transfer learning framework to transfer 3D and 2D features into a common sparse feature space, and hence it is preferable that both of them have similar logical meanings. Therefore, any useful information on the 3D and

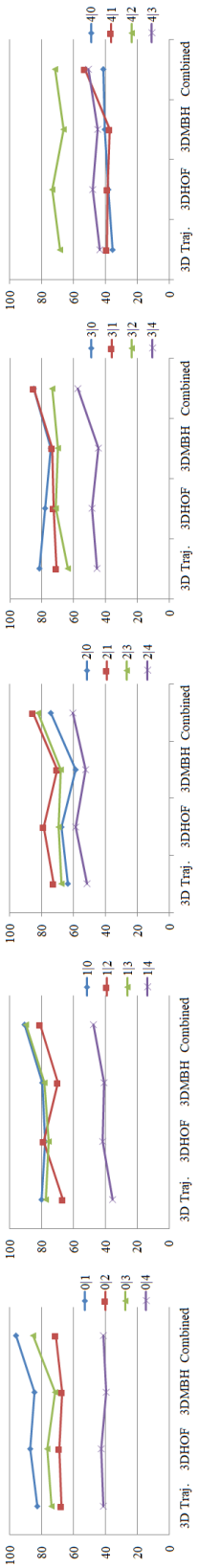


Fig. 4.22 Feature evaluation on IXMAS dataset according to different view transfer pairs.

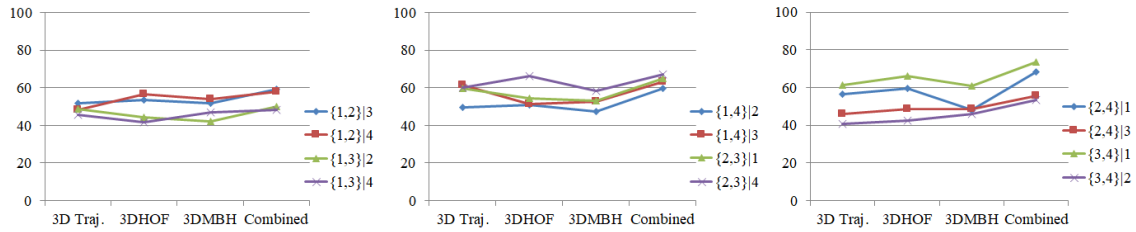


Fig. 4.23 Feature evaluation on UWA3DII dataset according to different view transfer pairs.

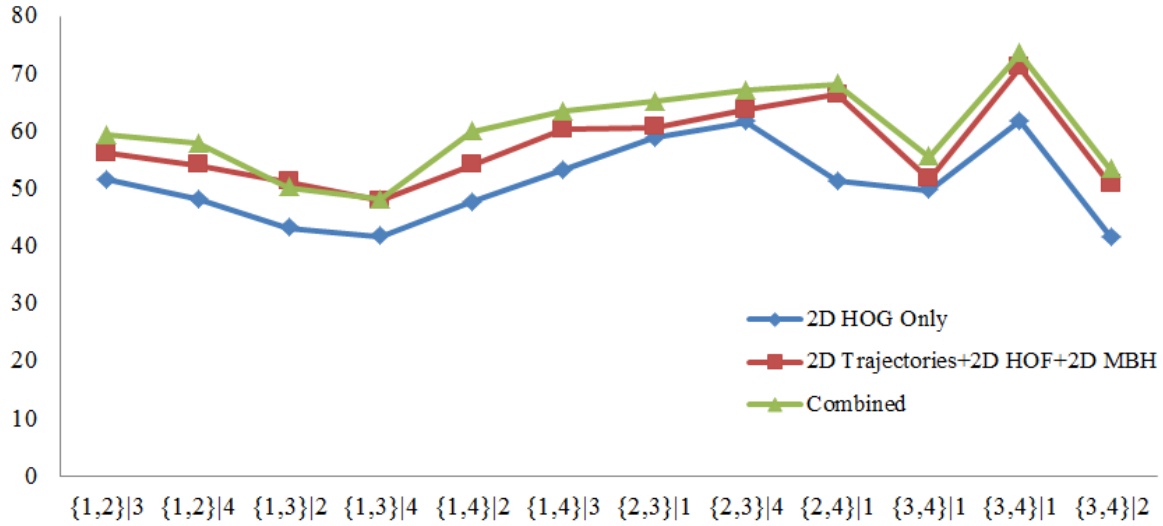


Fig. 4.24 2D HOG evaluation of the UWA3DII dataset according to different view transfer pairs.

2D action videos such as appearance will assist our system. The reason we do not propose 3DHOG is that the surface texture of a 3D model remains unchanged over time. We conduct an experiment on the UWA3DII dataset to show the importance of appearance feature 2D HOG.

Fig. 4.24 shows the performance on only, without and with using 2D HOG respectively. Features combined with 2D Trajectories, 2D HOF and 2D MBH perform better than only using 2D HOG in all the view transfer pairs. Because the movement related descriptors contain more view-invariant information than appearance related descriptors on the 2D action videos. Combined features also perform better than features without using 2D HOG, which shows the assistance of appearance information to our system.

4.6 Conclusions

In this chapter, we have proposed a view-invariant human action recognition framework. Unlike previous work, we construct a synthetic 3D and 2D video database using realistic human models, which is used to obtain the view-invariance through transfer dictionary learning. The trained dictionary is used to project real world 2D video into a view-invariant sparse representation, facilitating an arbitrary view action classifier. The use of synthetic data for initial training reduces the need for carefully captured video with view information. The synthetic dataset created in this project is open to the public, it is the first structured action dataset built with realistic human models for classification purposes. To enhance the quality of 3D motion description, we propose a new set of features known as the 3D dense trajectories, which consists of 3D trajectories, 3DHOF and 3DMBH. These features are complementary to each other and the combined feature set is highly effective for action classification. We demonstrate superior results in comparison to existing works in the IXMAS, NUCLA, UWA3DII and i3DPOST datasets.

In our system, we project the 3D and 2D videos into a common view-invariant sparse representation with the 3D and 2D dictionaries respectively. Theoretically speaking, it is possible to learn a dictionary that directly projects 2D video into 3D space, and consider the 3D space to be view-invariant. However, this is not practically possible. This is because 2D to 3D projection requires information that is not available in the 2D video. Even if a project matrix can be trained, the projected results will suffer from a large reconstruction error. In this research, we solve this problem by extracting the common view-invariant features in the 3D and 2D videos instead.

A main advantage of our framework is that the view-invariance transfer dictionary is pre-trained with a full synthetic dataset and fine-tuned with a small amount of real data. It is possible to include a large number of views in the synthetic dataset to learn a better view-invariant representation, even if the real data does not cover all of these views. Also, it is possible to introduce variations within each action class using computer graphics techniques such as motion style transfer to improve the richness of the dataset, which can enhance the classification accuracy. While existing work requires encoding and pooling parts to aggregate the local features, we use bag-of-words to effectively aggregate the local trajectories based features, motivated by the promising results from [40–43]. Specifically, we train a dictionary by using K-means to cluster the local features (e.g. HOG, HOF) into some visual words and then encode these local features by counting the occurrence of different visual words.

During the implementation, we found that the quality of the synthetic video could affect the classification accuracy of the system. This was the main motivation for us to utilize high-quality human models instead of simplified cylinder-based models as in previous

works. In the future, we are interested to explore if more realistic rendering (such as photorealistic rendering with global illuminations) and more realistic character movement (such as introducing secondary deformation to simulate the involuntary movement of body fat and clothings) would further improve the system performance.

In many datasets, the facing angles of the actors are not aligned with that camera viewpoints. As a result, the same action may appear differently for the same viewpoints dependent on the faced direction. As a future direction, we are interested in introducing the facing angle into the classification framework, such that the system can understand how the action may appear dependent on the orientation of the actor. Furthermore, when creating synthetic 2D videos, our current system samples projection viewpoints uniformly. With the facing angle, we may explore an optimal way of projection sampling that can optimize classification accuracy with a minimal number of synthetic 2D views.

Dictionary learning can be considered as a linear projection algorithm and can be limited in representing the view-invariance of 2D and 3D videos. In the future, we are interested in applying non-linear algorithms such as Neural Networks with synthetic training data to achieve better results. The potential challenges in using Neural Networks to learn the complex view-invariance is the need to tune a large number of hyper-parameters, as well as the need to design an optimal network architecture.

Chapter 5

Deep Feature for 3D Human Shape Reconstruction

With the increasing popularity of shooting equipment (e.g. mobile phones), the image becomes the most widely used content carrier. However, the content given by the image is very limited due to the lack of temporal information compared to the videos used in Chapter 4. Understanding human motion from a single image is a more challenging problem. The estimation of 3D human pose from a single image is a longstanding problem with many applications. Most previous approaches focus only on pose but ignore 3D human shape. In this chapter, we provide a solution that is fully automatic and estimates a 3D point cloud capturing human shape from a 2D image.

Automated 3D human body shape estimation has become a fundamental part of many practical applications. In this chapter, we seek to address the problem of reconstructing a 3D human shape from a 2D image. In this context, the major challenge faced is that the human form is defined by both the posture and the body dimensions. Additionally, the shape deformation is highly non-linear and therefore the shape surfaces of different postures are not easily comparable. Existing work typically relies on strong prior knowledge of human shapes to facilitate effective reconstruction, but such prior knowledge limits the representation, making the system unable to effectively represent fine details (e.g. fingers) or topological differences (e.g. clothing). In this chapter, we propose a novel end-to-end deep learning framework, capable of 3D human shape reconstruction from a 2D image without the need of a 3D prior parametric model. We employ a "prior-less" representation of the human shape using unordered point clouds. Due to the lack of prior information, comparing the generated and ground truth point clouds to evaluate the reconstruction error is challenging. We solve this problem by proposing an Earth Mover's Distance (EMD) function to find the optimal mapping between point clouds. Our experimental results show that we are able to obtain

a visually accurate estimation of the 3D human shape from a single 2D image, with some inaccuracy for heavily occluded body parts.

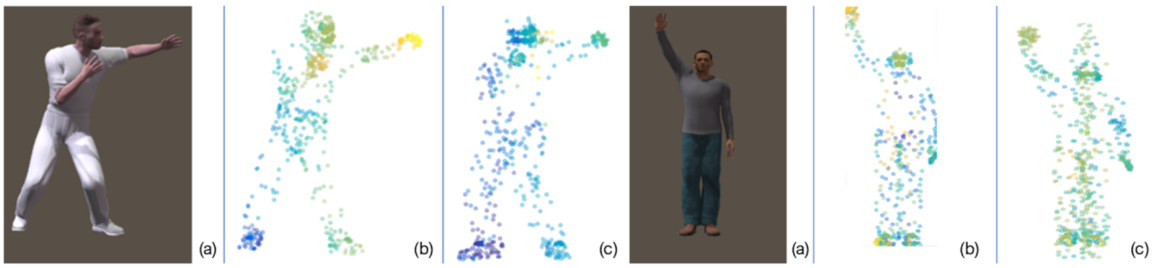


Fig. 5.1 Two examples of reconstruction: (a) 2D images, (b) ground truth 3D point cloud, and (c) reconstructed 3D point cloud.

5.1 Introduction

In this chapter, we tackle the problem of 2D to 3D reconstruction. We consider the use of 2D input as it is widely available and can be easily obtained using consumer hardware, as opposed to RGB-D images and 3D scans. Previous work in this field typically makes use of strong prior knowledge of plausible 3D human shapes. Two main approaches are parametric models [91] and indexed models [102]. The former represents the high-dimensional 3D model based upon low-dimensional parameters. The latter utilizes points with unique identities to represent landmarks on the body surface. While these method are effective in human shape reconstruction, they limit the representation capacity of the system. It becomes ineffective to represent shapes with large variations, such as the difference in body shape between male and female. It is also difficult to represent human surfaces with different topology, such as in the case of clothing or even physical disabilities.

To address this problem, rather than adopting the parametric model approach, we propose that the shape can be represented by directly generating a prior-less unordered point cloud, which is able to retain significantly more detail. Whilst the use of unordered point clouds can present a technical challenge, it can also prove advantageous over other approaches. One such advantage is that, unlike 2D-based representations such as images, there are no topological constraints upon the represented 3D shapes, meaning that direct correlation between points is not required. Also, when compared with 3D-volumetric grids, the point cloud set has higher efficiency by encoding only the points on the surface.

In order to facilitate the use of an unordered point cloud, we need to be able to measure the distance between the reconstructed points and the ground truths. Motivated by previous success on mesh processing [142, 143], we consider this as a transportation problem, and

propose a novel loss function that incorporates the Earth Mover's Distance. Such a method effectively determines the optimal alignment between two point cloud distributions, and allows for evaluation of the reconstruction accuracy for back-propagation.

In this chapter, we therefore present the following contributions:

- We propose an end-to-end deep learning framework for 3D human shape reconstruction from a 2D image without the need for a 3D prior parametric model.
- We propose a novel loss function based upon the Earth Mover's Distance [143] to evaluate the distances between unordered 3D point clouds representing human body shapes.

Our preliminary experimental results suggest that we are able to obtain a visually accurate estimation of the 3D human shape from a single 2D image, as shown in Fig.5.1. However it is also evident that self-occlusion has an impact upon the quality of the final output.

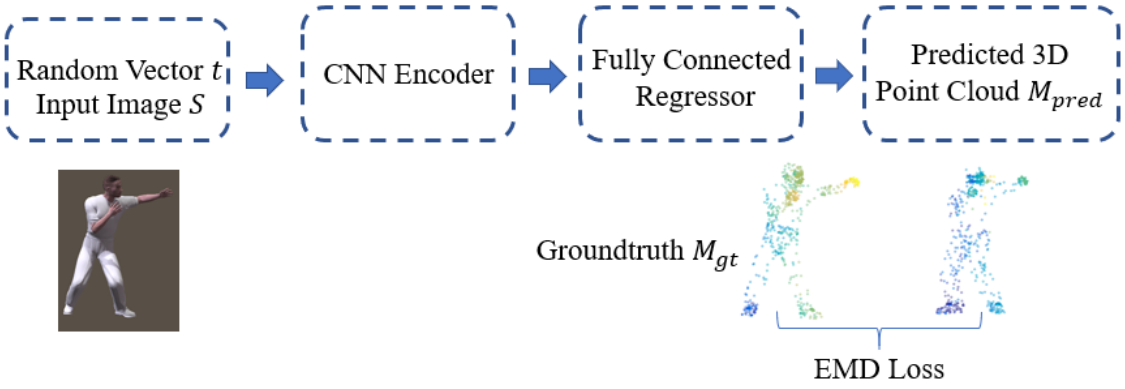


Fig. 5.2 The overview of our architecture.

5.2 Synthesising Training Data

Given the difficulty in obtaining suitably annotated 2D images with 3D ground truth data, and inspired by the success of [3], we synthesized a high-quality, realistic training dataset with which to train our system. To generate this dataset, we utilized skeletal human motion capture data from the CMU motion database [144]. We then utilized the Poser [145] software suite to obtain high-quality character meshes. We then applied skinning, to bind the skeleton to the mesh, in order to drive mesh deformation using the associated skeletal movement.

The generated frames of the meshes were then stored in a pairwise representation of 3D point clouds and 2D images. To obtain the 3D point clouds, each model was then projected onto a 2D image with a blank background. Each 3D human model, was normalized to 800 points and aligned using the Iterative Closest Point (ICP) algorithm. To generate 2D images, we projected the mesh with a predefined observation angle and with a simple local lighting model.

In our implementation, we selected ten diverse human models from the database, encompassing different genders and body sizes. As the process is automatic, we are easily able to enrich the database with more models and to incorporate other lighting strategies or more complex backgrounds. Figure 5.1 includes some examples of the 2D images (a) and the ground truth point clouds (b) generated.

5.3 EMD-informed CNN for Human Shape Reconstruction

To train a model that does not rely on prior knowledge, we represent the 3D human body shape as an unordered list of points on the human surface. The resultant system, therefore, can represent human body shapes with significant differences, even topologically. However, This also poses significant challenges in the 2D to 3D reconstruction process. We adapt a CNN-based network architecture, and devise a novel loss function informed by EMD to tackle this problem.

5.3.1 The Reconstruction Network

We propose a deep architecture that has strong representation ability and makes use of the statistics learned from the associated geometric data. An overview of the network architecture is shown in Fig. 5.2. Given an input image S and a random vector t, the network reconstructs a 3D point cloud M_r through an CNN encoder and a fully-connected regressor.

To model the uncertainty of the input image, we propose the incorporation of a random perturbation vector t as a part of the input, together with the input image S, as suggested by [146]. The inclusion of a small random vector allows the trained system to be more robust against noise. Also, once trained, we can change the random vector such that the system will generate different reconstruction results, which is useful in evaluating the system robustness.

The core of the network consists of a CNN encoder and a fully-connected regressor. The former is able to understand the features of images, while the latter can capture complex structures to generate the corresponding 3D point cloud. As we do not enforce prior knowledge, we use an unordered point cloud set $M = (x_i, y_i, z_i)_{i=1}^{N}$ to represent the 3D shapes,

where N is a predefined constant that represents the number of points in the point cloud. In our implementation, we set $N = 800$ as we have found that this value is generally sufficient to preserve the major structures of the body parts.

We define the ground truth as a probability distribution $P(\cdot|S)$ over the shapes conditioned on the input 2D image S to model the uncertainty from 2D to 3D. We train a deep neural network G as a conditional sampler from $P(\cdot|S)$:

$$M = G(S,t;\theta), \tag{5.1}$$

where θ denotes the network parameter, and $t \sim N(0,I)$ is the aforementioned random vector used to perturb the input. During testing, multiple samples of t are used to generate different predictions.

As for the implementation details, the encoder is composed of a combination of ReLU and convolution layers. It maps a random vector t and the input image S into a subspace. By using MoN (min of N) to model the uncertainty, the network can change its prediction based upon different random vectors. The regressor generates the 3D shape as an $N \times 3$ matrix, where each row represents the coordinates of one vertex.

5.3.2 The EMD-based Loss Function

While the use of an unordered point cloud frees the system from relying upon any priors, it is challenging to compare two unordered point clouds due to the lack of correspondence. Such a comparison is required when we build the reconstruction loss function. Motivated by the successes in applying Earth Mover's Function (EMD) to evaluate mesh distances [142, 143], we propose the use of EMD in our deep learning loss.

EMD evaluates the minimum overall distance between two point clouds by finding the optimal mapping between them. It optimizes a set of unidirectional flows to map the points. As EMD itself is an optimization problem, it is differentiable for point locations everywhere and therefore is an excellent choice to add to the deep learning architecture. It can also be calculated reasonably efficiently, which facilitates back-propagation of the data.

The loss function is defined as:

$$L(M_r, M_{gt}) = d_{EMD}(M_r, M_{gt}), \tag{5.2}$$

where M_r is the reconstructed 3D human shape, M_{gt} is the ground truth of each sample, d_{EMD} is the EMD calculated as:

$$d_{EMD}(M_1, M_2) = \min_{\phi: M_1 \to M_2} \sum_{x \in M_1, y \in M_2} ||y - \phi(x)||_2, \tag{5.3}$$

where $M_1, M_2 \in R^3$ has equal size, $m = |M1| = |M2|$ and $\phi : M_1 \to M_2$ is a bijection (i.e. flows), $||\quad||_2$ represents the root mean square point to point distance.

With the EMD-based loss function, the system can effectively evaluate the distance between the synthesized human shape and the ground-truth one for backpropagation during training.

5.4 Experimental Results

In this section, we demonstrate some preliminary experimental results generated by our system.

Fig.5.1 shows the point cloud reconstructed by our system compared with the ground truth based on a 2D image. The color visualizes the depth of the points in scale from blue (furthest away) to yellow (closest). Both examples show that our reconstructed point cloud resembles the body shape with the correct posture. For the left example, there are some errors around the right foot, where the foot direction is not pointing towards the viewer, and the hands are shown as being slightly further away. For the right example, the models' right hand appears lower and slightly further away, also there is some confusion with he distribution around the head and the thighs.

With the random vector, the same input image can have multiple plausible 3D shapes. Fig. 5.3 shows some differing 3D shapes produced using the same input image. They are largely consistent, which demonstrates the robustness of our system, and that the architecture is able to model the uncertainty of the predicted shape and the ambiguity of the input image. The top image shows a male model performing a throwing motion. There are small variations regarding the rotation of the lower part of the bodies, but the system successfully reconstructs the depth (i.e. the left leg is shown behind the right leg, based on the blue to yellow depth visualization scale). The bottom image shows a female model performing a low-kicking motion. There are some variations around the volume of the chest area, but the depth information is generally consistent and the overall posture provided is a good representation.

We also try to evaluate that impact that occlusion has on our system, which is one of the most challenging aspects in 2D to 3D reconstruction. Fig. 5.4 shows two example models from three different angles that generate different amounts of self-occlusion. With

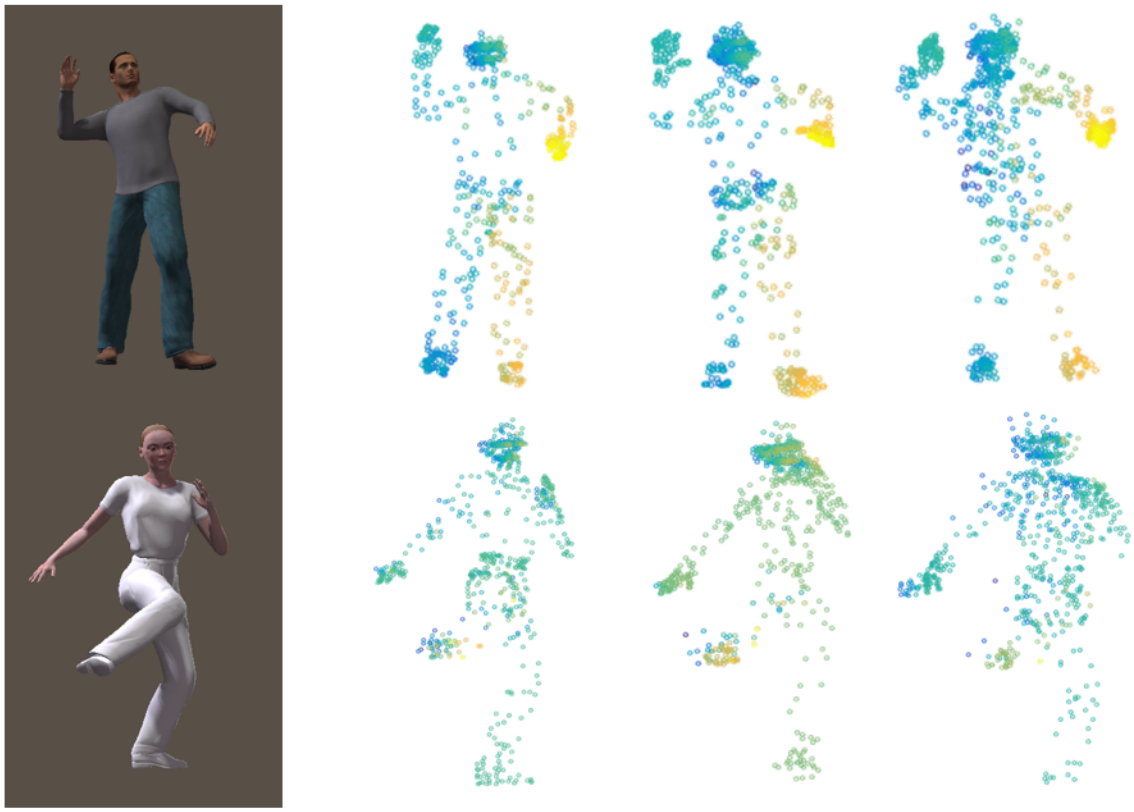


Fig. 5.3 Different possible shapes for the same image.

minimal occlusion, the system reconstructs the shape accurately. The accuracy decreases as the amount of occlusion increases. For the left example, which is a male character carrying out a walking motion, the lower body has more and more occlusion as we progress from the side view to the front view. As a result, the reconstruction quality degrades in these areas. The right example is a a female character performing a kicking motion. We observe similar problems around the upper body. Fig. (c) of the right example is an extreme case in which the whole right leg is occluded, this results in an unsatisfactory reconstruction. We also observe that the presence of occluded body parts may affect other body parts which are not occluded. This is likely due to the EMD function attempting to find the optimal mapping for the whole body, as such the occluded part confuses the mapping system.

5.5 Conclusion and Discussions

In this chapter, we propose an EMD-informed CNN framework for 2D to 3D point cloud reconstruction. Unlike the majority of the previous work, we experiment with a setup in

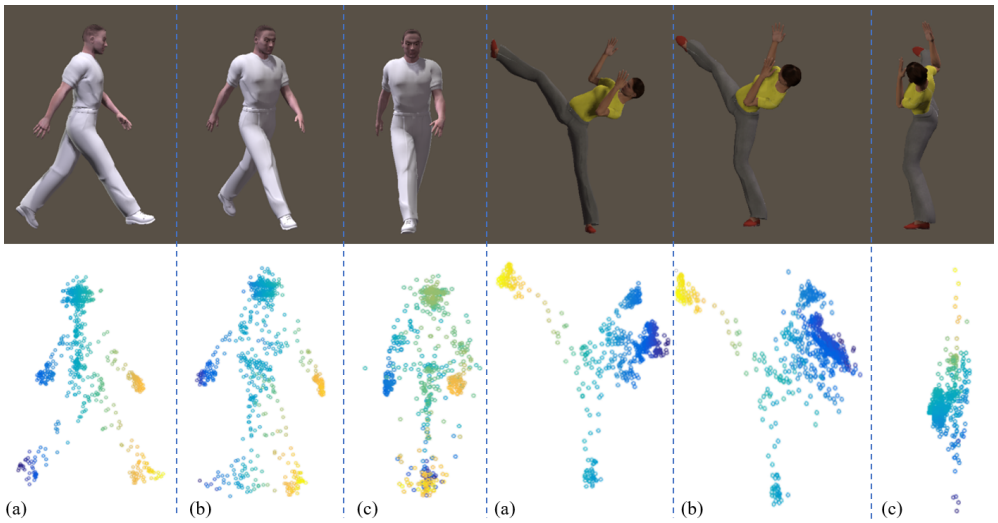


Fig. 5.4 The impact of occlusion in reconstruction: (a) minimal occlusion, (b) moderate occlusion, and (c) heavy occlusion.

which there is no prior-knowledge. Our EMD function successfully solves the problem of using an unordered point cloud for prior-less human shape representation. Furthermore, to enable sufficient high-quality training data, we employ a computer graphics pipeline to generate synthetic training data. Our method is able to solve the ambiguity of geometry in 2D images, however it suffers in situations where the model is heavily self-occluded.

Our preliminary results suggest that the generative ability of our network is good, and that it is able to model the uncertainty of predicted 3D shapes with a random vector. However, our system still suffers from a lack of discriminative ability and the generated 3D shapes are sometimes not realistic.

The use of EMD demonstrates high potential in matching two prior-less point clouds in order to evaluate the reconstruction loss. However, since the point to point distance is calculated in using Cartesian coordinates, when multiple joints are close together or occluded, the system has problems identifying which body parts the points should belong to, which results in poor quality during occlusion. Our future direction is to consider other coordinate systems within EMD, such at the Laplacian coordinate that can better represent surfaces, or the topology coordinate [147], that can effectively represent structures.

In our future work we also intend to explore the potential of incorporating Generative Adversary Networks (GANs) due to their strong discriminative ability [148]. Given its popularity in 3D reconstruction, we may also look to integrate a re-projection loss in our system by building a dual network which can project the reconstructed 3D model back to 2D, and minimise the difference between the re-projected 2D image and the input 2D image.

Finally, we have only tested with synthetic data. Whilst the images are almost photorealistic, in the future, we would like to test with real-world images, in the wild. We anticipate that this will be a challenge due to the high-frequency textures in real-world images, as well as the more complicated lighting conditions.

Chapter 6

Conclusions and Future work

In this chapter, the contributions of this thesis have been briefly concluded. Furthermore, the possible future work have been also discussed.

6.1 Conclusions

2D videos and images are the most popular medium to record human motion. However, 2D videos and images are not able to capture full 3D geometric information because of the limitation of single shooting angle. 2D appearance dramatically changes with the viewpoint changing. In this thesis, we try to investigate how to compensate for the lack of full geometric information in 2D based applications with view-invariance learnt from 3D models. We successfully extract the view-invariant features from high-quality 3D models to solve arbitrary view action recognition and image-based 3D human shape reconstruction problems.

We first propose an automatic gait analysis framework for musculoskeletal and neurological disorder diagnosis directly on 3D skeletal data. 3D skeletal data can record complete human movement information. Our system allows the machine to extract 3D movement features for disorders classification automatically. It can be enhanced by introducing prior medical knowledge to the existing features. We prove that a combination of human expertise and machine understanding would give a better description of these problems.

Unlike skeletal data, 2D videos and images are not able to capture full 3D geometric information. Therefore, we propose a view-invariant human action recognition framework. We construct a synthetic 3D and 2D video database using realistic human models, which is used to obtain the view-invariance through transfer dictionary learning. The trained dictionary is used to project real-world 2D video into a view-invariant sparse representation, facilitating an arbitrary view action classifier. The use of synthetic data for initial training reduces the

need for carefully captured video with view information. The synthetic dataset created in this project is open to the public, it is the first structured action dataset built with realistic human models for classification purposes. To enhance the quality of 3D motion description, we propose a new set of features known as the 3D dense trajectories, which consists of 3D trajectories, 3DHOF and 3DMBH. These features are complementary to each other, and the combined feature set is highly effective for action classification. We demonstrate superior results in comparison to existing works in the IXMAS, NUCLA, UWA3DII and i3DPOST datasets.

Furthermore, without temporal information, 3D human shape reconstruction from a single image is a highly under-determined problem, requiring strong prior knowledge of plausible 3D human shapes. We propose an end-to-end deep learning framework for 3D human shape reconstruction from a 2D image without any 3D parametric model. Going beyond skeletons, we propose a 3D point cloud based aligned method and a loss function to calculate the distance between 3D human body shapes so that the complex 3D models can easily fit into deep architecture.

6.2 Summary of Contributions

In this section, we summarize the contributions in this thesis.

- We propose an automatic framework for identifying musculoskeletal and neurological disorders among older people based on 3D skeletal data.
- We propose two new features called the 3D Relative Joints Displacement (3DRJDP) and the 6D Symmetric Relative Joint Displacement (6DSymRJDP) to capture the relationship of joint pairs across frames.
- We adapt feature selection methods including F-score, Neighborhood Component Analysis and ReliefF, for choosing an optimal feature set from the input features to optimize classification accuracy.
- We construct an openly accessible, comprehensive 3D gait database with the anonymised medical history of the subjects. The subjects are diagnosed as healthy, muscle weaknesses, joint problems and neurological defects by 3 medical doctors.
- We propose a new transfer dictionary learning framework that utilizes synthetic 2D and 3D training videos generated from realistic human models to learn a dictionary that can project a real world 2D video into a view-invariant sparse representation, which allows us to train an action classifier that works in an arbitrary view.

- We release our synthetic 2D and 3D dataset for public usage. This is the first structured action dataset built with realistic human models for high-quality action classification.
- We propose a new 3D feature set called the 3D dense trajectories consisting of 3D trajectories, 3DHOF and 3DMBH for a better description of motion in 3D. This can be considered as a 3D counterpart of the popular 2D feature dense trajectories [51].
- We propose an end-to-end deep learning framework for 3D human shape reconstruction from 2D image without any 3D parametric model.
- Going beyond skeletons, we propose a 3D point cloud based loss function to calculate the distance between 3D human body shapes, so that the complex 3D models can easily fit into deep architectures.

6.3 Discussion and Future Work

In this thesis, in order to solve the ambiguity in 2D based computer vision problems, we have tried to transfer the view-invariance from 3D models to 2D images or videos. There are two major sub-problems when we solve these 3D-2D applications: (1) How to describe the 2D images or videos and the 3D model? (2) How to establish a connection between 2D images and 3D models to make the transfer process smooth and efficient? For the first problem, we find that using equivalent representation methods for 2D videos and 3D models (e.g. 2D trajectories and 3D trajectories) is easier to be integrated into the architecture and is able to improve the efficiency of the system. Conversely, if the representation of both ends are not equivalent (e,g. raw image and point cloud data), we have to design more complex architecture to make up for this problem. For the second problem, we may have three potential way to improve the system performance in the future:

Non-linear Transfer Learning Framework In Chapter 4, we use transfer dictionary learning to bridge the connection between 2D and 3D. This method is proved to be efficient. However, dictionary learning method learns a linear mapping between 2D and 3D data, which can not precisely describe the non-linear projection process between 3D and 2D data. Therefore, some non-linear algorithms are worth trying to explore whether it is suitable to model the relationship between the 3D model and its multiple 2D projections.

Discriminativeness Improvement via Generative Adversary Networks (GAN) Generative Adversary Networks (GAN) has been widely used in today's research due to its strong discriminative ability [148]. In Chapter 5, our architecture has good generative ability but lack the discriminative ability. Especially when the input 2D image suffers from severe

self-occlusion, the predicted 3D shape becomes different to identify. Therefore, we will try to add an adversarial loss to the original EMD based loss function. We will build a database of healthy 3D shapes. If the predicted 3D shape is significantly different from any health shapes in the database. The system will reject it and find the nearest healthy shape in the database to improve the outcome.

Dual Network Re-projection loss is a very popular tool in the 3D reconstruction area. The idea is simple [93]. They project the reconstructed 3D model back to 2D and minimise the difference between the re-projected 2D image with the input 2D image. We can also integrate this re-projection loss in our system to build a dual network. The problem will be how to compute the difference between the image and the projected point cloud. Using silhouette may be an intermediate method to balance these two types of images.

References

[1] Ankur Gupta, Julieta Martinez, James J Little, and Robert J Woodham. 3D pose from motion for cross-view action recognition via non-linear circulant temporal encoding. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2601–2608, 2014.

[2] Hossein Rahmani and Ajmal Mian. Learning a non-linear knowledge transfer model for cross-view action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2458–2466, 2015.

[3] Jingtian Zhang, Lining Zhang, Hubert PH Shum, and Ling Shao. Arbitrary view action recognition via transfer dictionary learning on synthetic training data. In *IEEE International Conference on Robotics and Automation*, pages 1678–1684. IEEE, 2016.

[4] Pingkun Yan, Saad M Khan, and Mubarak Shah. Learning 4d action feature models for arbitrary view action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–7. IEEE, 2008.

[5] Beom-Chan Lee, Alberto Fung, and Timothy A Thrasher. The effects of coding schemes on vibrotactile biofeedback for dynamic balance training in parkinson's disease and healthy elderly individuals. *IEEE Transaction on Neural Systems and Rehabilitation Engineering*, 26(1):153–160, 2018.

[6] Centers for Disease Control, Prevention, et al. Prevalence and most common causes of disability among adults–united states, 2005. *MMWR: Morbidity and Mortality Weekly Report*, 58(16):421–426, 2009.

[7] Joe Verghese, Aaron LeValley, Charles B Hall, Mindy J Katz, Anne F Ambrose, and Richard B Lipton. Epidemiology of gait disorders in community-residing older adults. *Journal of the American Geriatrics Society*, 54(2):255–261, 2006.

[8] Neil B Alexander. Gait disorders in older adults. *Journal of the American Geriatrics Society*, 44(4):434–451, 1996.

[9] Karen M Ostrosky, Jessie M VanSwearingen, Ray G Burdett, and Zena Gee. A comparison of gait characteristics in young and old subjects. *Physical Therapy*, 74(7):637–644, 1994.

[10] Lily Lee and W Eric L Grimson. Gait analysis for recognition and classification. In *IEEE International Conference on Automatic Face and Gesture Recognition*, pages 155–162. IEEE, 2002.

[11] Tuan D Pham. Texture classification and visualization of time series of gait dynamics in patients with neuro-degenerative diseases. *IEEE Transaction on Neural Systems and Rehabilitation Engineering*, 26(1):188–196, 2018.

[12] Stefan H Holzreiter and Monika E Köhle. Assessment of gait patterns using neural networks. *Journal of Biomechanics*, 26(6):645–651, 1993.

[13] JG Barton and A Lees. An application of neural networks for distinguishing gait patterns on the basis of hip-knee joint angle diagrams. *Gait & Posture*, 5(1):28–33, 1997.

[14] Rezaul K Begg, Marimuthu Palaniswami, and Brendan Owen. Support vector machines for automated gait classification. *IEEE Transaction on Biomedical Engineering*, 52(5):828–838, 2005.

[15] Ahsan H Khandoker, Daniel TH Lai, Rezaul K Begg, and Marimuthu Palaniswami. Wavelet-based feature extraction for support vector machines for screening balance impairments in the elderly. *IEEE Transaction on Neural Systems and Rehabilitation Engineering*, 15(4):587–597, 2007.

[16] Richard Baker. Gait analysis methods in rehabilitation. *Journal of Neuroengineering and Rehabilitation*, 3(1):4, 2006.

[17] Tuomas Liikavainio, Juha Isolehto, Heikki J Helminen, Jarmo Perttunen, Vesa Lepola, Ilkka Kiviranta, Jari PA Arokoski, and Paavo V Komi. Loading and gait symmetry during level and stair walking in asymptomatic subjects with knee osteoarthritis: importance of quadriceps femoris in reducing impact force during heel strike? *The Knee*, 14(3):231–238, 2007.

[18] Kara K Patterson, William H Gage, Dina Brooks, Sandra E Black, and William E McIlroy. Evaluation of gait symmetry after stroke: a comparison of current methods and recommendations for standardization. *Gait & Posture*, 31(2):241–246, 2010.

[19] Ryan P Hubble, Geraldine A Naughton, Peter A Silburn, and Michael H Cole. Wearable sensor use for assessing standing balance and walking stability in people with parkinson's disease: a systematic review. *PloS One*, 10(4):e0123705, 2015.

[20] Ioannis Papavasileiou, Wenlong Zhang, Xin Wang, Jinbo Bi, Li Zhang, and Song Han. Classification of neurological gait disorders using multi-task feature learning. In *International Conference on Connected Health: Applications, Systems and Engineering Technologies*, pages 195–204. IEEE, 2017.

[21] Anil Jain and Douglas Zongker. Feature selection: Evaluation, application, and small sample performance. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 19(2):153–158, 1997.

[22] Avrim L Blum and Pat Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97(1):245–271, 1997.

[23] Bir Bhanu and Ju Han. Human recognition on combining kinematic and stationary features. In *Audio- and Video-Based Biometric Person Authentication*, pages 600–608. Springer, 2003.

[24] Mark S Nixon, Tieniu Tan, and Rama Chellappa. *Human identification based on gait*, volume 4. Springer Science & Business Media, 2010.

[25] Yiwei Chen and Chihjen Lin. Combining svms with various feature selection strategies. *Feature Extraction*, pages 315–324, 2006.

[26] Marko Robnik-Šikonja and Igor Kononenko. Theoretical and empirical analysis of relieff and rrelieff. *Machine Learning*, 53(1-2):23–69, 2003.

[27] Wei Yang, Kuanquan Wang, and Wangmeng Zuo. Neighborhood component feature selection for high-dimensional data. *Journal of Computers*, 7(1):161–168, 2012.

[28] Sara Mulroy, JoAnne Gronley, Walt Weiss, Craig Newsam, and Jacquelin Perry. Use of cluster analysis for gait pattern classification of patients in the early and late recovery phases following stroke. *Gait & Posture*, 18(1):114–125, 2003.

[29] Sharon Kinsella and Kieran Moran. Gait pattern categorization of stroke participants with equinus deformity of the foot. *Gait & Posture*, 27(1):144–151, 2008.

[30] Katarzyna Kaczmarczyk, Andrzej Wit, Maciej Krawczyk, and Jacek Zaborski. Gait classification in post-stroke patients using artificial neural networks. *Gait & Posture*, 30(2):207–210, 2009.

[31] Nooritawati Md Tahir and Hany Hazfiza Manap. Parkinson disease gait classification based on machine learning approach. *Journal of Applied Sciences*, 12(2):180–185, 2012.

[32] Sang-Hong Lee and Joon S Lim. Parkinson's disease classification using gait characteristics and wavelet-based feature extraction. *Expert Systems with Applications*, 39(8):7338–7344, 2012.

[33] Moshe Blank, Lena Gorelick, Eli Shechtman, Michal Irani, and Ronen Basri. Actions as space-time shapes. In *IEEE International Conference on Computer Vision*, volume 2, pages 1395–1402. IEEE, 2005.

[34] Piotr Dollár, Vincent Rabaud, Garrison Cottrell, and Serge Belongie. Behavior recognition via sparse spatio-temporal features. In *IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pages 65–72. IEEE, 2005.

[35] Ivan Laptev. On space-time interest points. *International Journal of Computer Vision*, 64(2-3):107–123, 2005.

[36] Jingen Liu and Mubarak Shah. Learning human actions via information maximization. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[37] Jingen Liu, Yang Yang, and Mubarak Shah. Learning semantic visual vocabularies using diffusion distance. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 461–468. IEEE, 2009.

[38] Zhe Lin, Zhuolin Jiang, and Larry S Davis. Recognizing actions by shape-motion prototype trees. In *IEEE International Conference on Computer Vision*, pages 444–451. IEEE, 2009.

[39] Fengjun Lv and Ramakant Nevatia. Single view human action recognition using key pose matching and viterbi path searching. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

[40] Shandong Wu, Omar Oreifej, and Mubarak Shah. Action recognition in videos acquired by a moving camera using motion decomposition of lagrangian particle trajectories. In *IEEE International Conference on Computer Vision*, pages 1419–1426. IEEE, 2011.

[41] Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng Lin Liu. Dense trajectories and motion boundary descriptors for action recognition. *International Journal of Computer Vision*, 103(1):60–79, 2013.

[42] Heng Wang, Alexander Klaser, Cordelia Schmid, and Cheng Lin Liu. Action recognition by dense trajectories. In *IEEE International Conference on Computer Vision*, pages 3169–3176. IEEE, 2011.

[43] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *IEEE International Conference on Computer Vision*, pages 3551–3558, 2013.

[44] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *IEEE International Conference on Computer Vision*, pages 1725–1732. IEEE, 2014.

[45] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems*, pages 568–576, 2014.

[46] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3D convolutional networks. In *IEEE International Conference on Computer Vision*, pages 4489–4497. IEEE, 2015.

[47] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European Conference on Computer Vision*, pages 20–36. Springer, 2016.

[48] Adriana Kovashka and Kristen Grauman. Learning a hierarchy of discriminative space-time neighborhood features for human action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2046–2053. IEEE, 2010.

[49] Ali Farhadi and Mostafa Kamali Tabrizi. Learning to recognize activities from the wrong view point. In *European Conference on Computer Vision*, pages 154–166. Springer, 2008.

[50] Jingen Liu, Mubarak Shah, Benjamin Kuipers, and Silvio Savarese. Cross-view action recognition via view knowledge transfer. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3209–3216. IEEE, 2011.

[51] Jiang Wang, Xiaohan Nie, Yin Xia, Ying Wu, and Song-Chun Zhu. Cross-view action modeling, learning and recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2649–2656, 2014.

[52] Chun-Hao Huang, Yi-Ren Yeh, and Yu-Chiang Frank Wang. Recognizing actions across cameras by exploring the correlated subspace. In *European Conference on Computer Vision*, pages 342–351. Springer, 2012.

[53] Daniel Weinland, Edmond Boyer, and Remi Ronfard. Action recognition from arbitrary views using 3D exemplars. In *IEEE International Conference on Computer Vision*, pages 1–7. IEEE, 2007.

[54] Hossein Rahmani, Ajmal Mian, and Mubarak Shah. Learning a deep model for human action recognition from novel viewpoints. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[55] Mihael Ankerst, Gabi Kastenmüller, Hans-Peter Kriegel, and Thomas Seidl. 3D shape histograms for similarity search and classification in spatial databases. In *International Symposium on Spatial Databases*, pages 207–226. Springer, 1999.

[56] Marcel Körtgen, Gil-Joo Park, Marcin Novotni, and Reinhard Klein. 3D shape matching with 3D shape contexts. In *Central European Seminar on Computer Graphics*, volume 3, pages 5–17. Budmerice, 2003.

[57] Peng Huang and Adrian Hilton. Shape-colour histograms for matching 3D video sequences. In *International Conference on Computer Vision Workshops*, pages 1510–1517. IEEE, 2009.

[58] Selen Pehlivan and Pınar Duygulu. A new pose-based representation for recognizing actions from multiple cameras. *Computer Vision and Image Understanding*, 115(2):140–151, 2011.

[59] Daniel Weinland, Remi Ronfard, and Edmond Boyer. Free viewpoint action recognition using motion history volumes. *Computer Vision and Image Understanding*, 104(2):249–257, 2006.

[60] Massimiliano Pierobon, Marco Marcon, Augusto Sarti, and Stefano Tubaro. 3-d body posture tracking for human action template matching. In *International Conference on Acoustics, Speech and Signal Processing*, volume 2, pages II–II. IEEE, 2006.

[61] Isaac Cohen and Hongxia Li. Inference of human postures by classification of 3D human body shape. In *IEEE International Workshop on Analysis and Modeling of Faces and Gestures*, pages 74–81. IEEE, 2003.

[62] Baochang Zhang, Yun Yang, Chen Chen, Linlin Yang, Jungong Han, and Ling Shao. Action recognition using 3D histograms of texture and a multi-class boosting classifier. *IEEE Transactions on Image Processing*, 26(10):4648–4660, 2017.

[63] Mengyuan Liu, Hong Liu, Chen Chen, and Maryam Najafian. Energy-based global ternary image for action recognition using sole depth sequences. In *International Conference on 3D Vision*, pages 47–55. IEEE, 2016.

[64] Mengyuan Liu, Hong Liu, and Chen Chen. Enhanced skeleton visualization for view invariant human action recognition. *Pattern Recognition*, 68:346–362, 2017.

[65] Kangkan Wang, Guofeng Zhang, and Shihong Xia. Templateless non-rigid reconstruction and motion tracking with a single RGB-D camera. *IEEE Transactions on Image Processing*, 26(12):5966–5979, 2017.

[66] Chengcheng Jia and Yun Fu. Low-rank tensor subspace learning for RGB-D action recognition. *IEEE Transactions on Image Processing*, 25(10):4641–4652, 2016.

[67] Yu Kong and Yun Fu. Discriminative relational representation learning for RGB-D action recognition. *IEEE Transactions on Image Processing*, 25(6):2856–2865, 2016.

[68] Paul Scovanner, Saad Ali, and Mubarak Shah. A 3-dimensional sift descriptor and its application to action recognition. In *ACM international conference on Multimedia*, pages 357–360. ACM, 2007.

[69] Geert Willems, Tinne Tuytelaars, and Luc Van Gool. An efficient dense and scale-invariant spatio-temporal interest point detector. In *European Conference on Computer Vision*, pages 650–663. Springer, 2008.

[70] Ross Messing, Chris Pal, and Henry Kautz. Activity recognition using the velocity histories of tracked keypoints. In *IEEE International Conference on Computer Vision*, pages 104–111. IEEE, 2009.

[71] Pyry Matikainen, Martial Hebert, and Rahul Sukthankar. Trajectons: Action recognition through the motion analysis of tracked features. In *International Conference on Computer Vision Workshops*, pages 514–521. IEEE, 2009.

[72] Ju Sun, Xiao Wu, Shuicheng Yan, Loong-Fah Cheong, Tat-Seng Chua, and Jintao Li. Hierarchical spatio-temporal context modeling for action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2004–2011. IEEE, 2009.

[73] Ju Sun, Yadong Mu, Shuicheng Yan, and Loong-Fah Cheong. Activity recognition using dense long-duration trajectories. In *IEEE International Conference on Multimedia and Expo*, pages 322–327. IEEE, 2010.

[74] Fang Liu, Xiangmin Xu, Shuoyang Qiu, Chunmei Qing, and Dacheng Tao. Simple to complex transfer learning for action recognition. *IEEE Transactions on Image Processing*, 25(2):949–960, 2016.

[75] Tiantian Xu, Fan Zhu, Edward K Wong, and Yi Fang. Dual many-to-one-encoder-based transfer learning for cross-dataset human action recognition. *Image and Vision Computing*, 55:127–137, 2016.

[76] Mingyuan Zhou, Haojun Chen, Lu Ren, Guillermo Sapiro, Lawrence Carin, and John W Paisley. Non-parametric bayesian dictionary learning for sparse image representations. In *Advances in Neural Information Processing Systems*, pages 2295–2303, 2009.

[77] John Wright, Allen Y Yang, Arvind Ganesh, S Shankar Sastry, and Yi Ma. Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):210–227, 2009.

[78] Yang Wang and Greg Mori. Hidden part models for human action recognition: Probabilistic versus max margin. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(7):1310–1323, 2011.

[79] Michal Aharon, Michael Elad, and Alfred Bruckstein. *rmk*-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322, 2006.

[80] Qiang Qiu, Vishal M Patel, Pavan Turaga, and Rama Chellappa. Domain adaptive dictionary learning. In *European Conference on Computer Vision*, pages 631–645. Springer, 2012.

[81] Jingjing Zheng, Zhuolin Jiang, and Rama Chellappa. Cross-view action recognition via transferable dictionary learning. *IEEE Transactions on Image Processing*, 25(6):2542–2556, 2016.

[82] Jingtian Zhang, Hubert PH Shum, Jungong Han, and Ling Shao. Action recognition from arbitrary views using transferable dictionary learning. *IEEE Transactions on Image Processing*, 27(10):4709–4723, 2018.

[83] Nikolas Hesse, Sergi Pujades, Javier Romero, Michael J Black, Christoph Bodensteiner, Michael Arens, Ulrich G Hofmann, Uta Tacke, Mijna Hadders-Algra, Raphael Weinberger, et al. Learning an infant body model from RGB-D data for accurate full body motion analysis. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 792–800. Springer, 2018.

[84] Pengpeng Hu, Edmond SL Ho, Nauman Aslam, Taku Komura, and Hubert PH Shum. A new method to evaluate the dynamic air gap thickness and garment sliding of virtual clothes during walking. *Textile Research Journal*, page 0040517519826930, 2019.

[85] Peng Guan, Alexander Weiss, Alexandru O Balan, and Michael J Black. Estimating human shape and pose from a single image. In *IEEE International Conference on Computer Vision*, pages 1381–1388. IEEE, 2009.

[86] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. Scape: shape completion and animation of people. In *ACM Transactions on Graphics (TOG)*, volume 24, pages 408–416. ACM, 2005.

[87] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. In *ACM Transactions on Graphics (TOG)*, volume 23, pages 309–314. ACM, 2004.

[88] Leonid Sigal, Alexandru Balan, and Michael J Black. Combined discriminative and generative articulated pose and non-rigid shape estimation. In *Advances in Neural Information Processing Systems*, pages 1337–1344, 2008.

[89] Shizhe Zhou, Hongbo Fu, Ligang Liu, Daniel Cohen-Or, and Xiaoguang Han. Parametric reshaping of human bodies in images. In *ACM Transactions on Graphics (TOG)*, volume 29, page 126. ACM, 2010.

[90] Yu Chen, Tae-Kyun Kim, and Roberto Cipolla. Inferring 3D shapes and deformations from single views. In *European Conference on Computer Vision*, pages 300–313. Springer, 2010.

[91] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. *ACM Transactions on Graphics (TOG)*, 34(6):248, 2015.

[92] Timo von Marcard, Roberto Henschel, Michael J Black, Bodo Rosenhahn, and Gerard Pons-Moll. Recovering accurate 3D human pose in the wild using imus and a moving camera. In *European Conference on Computer Vision*, pages 601–617, 2018.

[93] Angjoo Kanazawa, Michael J Black, David W Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 7122–7131, 2018.

[94] Chao Zhang, Sergi Pujades, Michael J Black, and Gerard Pons-Moll. Detailed, accurate, human shape estimation from clothed 3D scan sequences. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4191–4200, 2017.

[95] Gul Varol, Duygu Ceylan, Bryan Russell, Jimei Yang, Ersin Yumer, Ivan Laptev, and Cordelia Schmid. Bodynet: Volumetric inference of 3D human body shapes. In *European Conference on Computer Vision*, pages 20–36, 2018.

[96] Tao Yu, Zerong Zheng, Kaiwen Guo, Jianhui Zhao, Qionghai Dai, Hao Li, Gerard Pons-Moll, and Yebin Liu. Doublefusion: Real-time capture of human performances with inner body shapes from a single depth sensor. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 7287–7296, 2018.

[97] Tejas D Kulkarni, Pushmeet Kohli, Joshua B Tenenbaum, and Vikash Mansinghka. Picture: A probabilistic programming language for scene perception. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4390–4399, 2015.

[98] Hsi-Jian Lee, CHEN Zen, et al. Determination of 3D human-body postures from a single view. *Computer Vision Graphics and Image Processing*, 30(2):148–168, 1985.

[99] Hashim Yasin, Umar Iqbal, Bjorn Kruger, Andreas Weber, and Juergen Gall. A dual-source approach for 3D pose estimation from a single image. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4948–4956, 2016.

[100] Catalin Ionescu, Joao Carreira, and Cristian Sminchisescu. Iterated second-order label sensitive pooling for 3D human pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1661–1668, 2014.

[101] Edgar Simo-Serra, Arnau Ramisa, Guillem Alenyà, Carme Torras, and Francesc Moreno-Noguer. Single image 3D human pose estimation from noisy observations. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2673–2680. IEEE, 2012.

[102] Christoph Lassner, Javier Romero, Martin Kiefel, Federica Bogo, Michael J Black, and Peter V Gehler. Unite the people: Closing the loop between 3D and 2D human representations. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 6050–6059, 2017.

[103] Gerard Pons-Moll, Sergi Pujades, Sonny Hu, and Michael J Black. Clothcap: Seamless 4d clothing capture and retargeting. *ACM Transactions on Graphics (TOG)*, 36(4):73, 2017.

[104] Jonathan Starck and Adrian Hilton. Surface capture for performance-based animation. *IEEE Computer Graphics and Applications*, 27(3):21–31, 2007.

[105] Gul Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J Black, Ivan Laptev, and Cordelia Schmid. Learning from synthetic humans. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 109–117, 2017.

[106] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3D classification and segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 652–660, 2017.

[107] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems*, pages 5099–5108, 2017.

[108] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3D object reconstruction from a single image. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 605–613, 2017.

[109] Tran Nhat Hung and Young Soo Suh. Inertial sensor-based two feet motion tracking for gait analysis. *Sensors*, 13(5):5614–5629, 2013.

[110] Orawan Kuha and Vanichvarotm Chitnapa Thamanavat Nuntasak Bunmeepipit, Boorinee. Comparative study of mini-mental state examination thai 2002 (mmse-thai 2002) and thai mini-mental state examination (tmse) in elderly screening test for cognitive impairment. *Journal of Gerontology and Geriatric Medicine*, 10(1):19–24, 2009.

[111] Ladda Thiamwong. Psychometric testing of the falls efficacy scale-international. *Songklanagarind Medical Journal*, 29(6):277–287, 2012.

[112] Worasak Rueangsirarak, Anthony S Atkins, Bernadette Sharp, Nopasit Chakpitak, Komsak Meksamoot, and Prapas Pothongsunun. Clustering the clusters–knowledge enhancing tool for diagnosing elderly falling risk. *International Journal of Healthcare Technology and Management*, 14(1-2):39–60, 2013.

[113] Kenta Takayasu, Kenji Yoshida, Takao Mishima, Masato Watanabe, Tadashi Matsuda, and Hidefumi Kinoshita. Analysis of the posture pattern during robotic simulator tasks using an optical motion capture system. *Surgical Endoscopy*, 32(1):183–190, 2018.

[114] Thomas D. Collins, Salim N. Ghoussayni, David J. Ewins, and Jenny A. Kent. A six degrees-of-freedom marker set for gait analysis: repeatability and comparison with a modified helen hayes set. *Gait & Posture*, 30(2):173–180, 2009.

[115] Karen J Mickle, Bridget J Munro, Stephen R Lord, Hylton B Menz, and Julie R Steele. Gait, balance and plantar pressures in older people with toe deformities. *Gait & Posture*, 34(3):347–351, 2011.

[116] Fei Han, Brian Reily, William Hoff, and Hao Zhang. Space-time representation of people based on 3D skeletal data: A review. *Computer Vision and Image Understanding*, 158:85–105, 2017.

[117] Jehee Lee and Sung Yong Shin. A hierarchical approach to interactive motion editing for human-like figures. In *Conference on Computer Graphics and Interactive Techniques*, pages 39–48. ACM Press/Addison-Wesley Publishing Co., 1999.

[118] Maddock Meredith, Steve Maddock, et al. Motion capture file formats explained. *Department of Computer Science, University of Sheffield*, 211:241–244, 2001.

[119] Washef Ahmed, Kunal Chanda, and Soma Mitra. Vision based hand gesture recognition using dynamic time warping for indian sign language. In *International Conference on Information Science*, pages 120–125. IEEE, 2016.

[120] Jacob Goldberger, Geoffrey E Hinton, Sam T. Roweis, and Ruslan R Salakhutdinov. Neighbourhood components analysis. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems*, pages 513–520. MIT Press, 2005.

[121] Kenji Kira and Larry A. Rendell. A practical approach to feature selection. In *Proceedings of the Ninth International Workshop on Machine Learning*, ML92, pages 249–256, San Francisco, CA, USA, 1992. Morgan Kaufmann Publishers Inc.

[122] Igor Kononenko, Marko Robnik-Sikonja, Marko Robnik, and Uros Pompe. Relieff for estimation and discretization of attributes in classification, regression, and ilp problems, 1996.

[123] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, pages 144–152, New York, NY, USA, 1992. ACM.

[124] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, September 1995.

[125] Stefan Knerr, Léon Personnaz, and Gérard Dreyfus. Single-layer learning revisited: a stepwise procedure for building and training a neural network. In *Neurocomputing*, pages 41–50. Springer, 1990.

[126] ChihChung Chang and ChihJen Lin. LIBSVM: A library for support vector machines. *ACM Transaction on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[127] Naoya Iwamoto, Hubert P. H. Shum, Longzhi Yang, and Shigeo Morishima. Multilayer lattice model for real-time dynamic character deformation. *Computer Graphic Forum*, 34(7):99–109, Oct 2015.

[128] Chris Hecker, Bernd Raabe, Ryan W Enslow, John DeWeese, Jordan Maynard, and Kees van Prooijen. Real-time motion retargeting to highly varied user-created morphologies. *ACM Transactions on Graphics (TOG)*, 27(3):27, 2008.

[129] Joëlle Tilmanne, Raphaël Sebbe, and Thierry Dutoit. A database for stylistic human gait modeling and synthesis. In *Workshop on Multimodal Interfaces, Paris, France*, pages 91–94. Citeseer, 2008.

[130] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online dictionary learning for sparse coding. In *International Conference on Machine Learning*, pages 689–696. ACM, 2009.

[131] Fan Zhu and Ling Shao. Weakly-supervised cross-domain dictionary learning for visual recognition. *International Journal of Computer Vision*, 109(1-2):42–59, 2014.

[132] Zhuolin Jiang, Zhe Lin, and Larry S Davis. Learning a discriminative dictionary for sparse coding via label consistent k-svd. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1697–1704. IEEE, 2011.

[133] Gene H Golub, Per Christian Hansen, and Dianne P O'Leary. Tikhonov regularization and total least squares. *SIAM Journal on Matrix Analysis and Applications*, 21(1):185–194, 1999.

[134] Ruonan Li and Todd Zickler. Discriminative virtual views for cross-view action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2855–2862. IEEE, 2012.

[135] Zhong Zhang, Chunheng Wang, Baihua Xiao, Wen Zhou, Shuang Liu, and Cunzhao Shi. Cross-view action recognition via a continuous virtual path. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2690–2697, 2013.

[136] Binlong Li, Octavia I Camps, and Mario Sznaier. Cross-view activity recognition using hankelets. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1362–1369. IEEE, 2012.

[137] Nikolaos Gkalelis, Hansung Kim, Adrian Hilton, Nikos Nikolaidis, and Ioannis Pitas. The i3dpost multi-view and 3D human action/interaction database. In *IEEE Conference on Visual Media Production*, pages 159–168. IEEE, 2009.

[138] Georgia Gkioxari and Jitendra Malik. Finding action tubes. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 759–768, 2015.

[139] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2625–2634, 2015.

[140] Michael B Holte, Thomas B Moeslund, Nikos Nikolaidis, and Ioannis Pitas. 3D human action recognition for multi-view camera systems. In *International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission*, pages 342–349. IEEE, 2011.

[141] Alexandros Iosifidis, Nikos Nikolaidis, and Ioannis Pitas. Movement recognition exploiting multi-view information. In *IEEE International Workshop on Multimedia Signal Processing*, pages 427–431. IEEE, 2010.

[142] Joseph Henry, Hubert PH Shum, and Taku Komura. Interactive formation control in complex environments. *IEEE Transactions on Visualization and Computer Graphics*, 20(2):211–222, 2013.

[143] Yijun Shen, Longzhi Yang, Edmond SL Ho, and Hubert PH Shum. Interaction-based human activity comparison. *IEEE Transactions on Visualization and Computer Graphics*, 2019.

[144] Ralph Gross and Jianbo Shi. *The CMU Motion of Body (MoBo) Database*, 2001.

[145] EunKyoung Yang and SookJin Kim. 3D character virtual costume making software usability assessment—focusing on poser 3D character virtual costume making. *Journal on Digital*, 14(1):863–876, 2014.

[146] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1765–1773, 2017.

[147] Ana-Sabina Irimia, Jacky CP Chan, Kamlesh Mistry, Wei Wei, and Edmond SL Ho. Emotion transfer for hand animation. In *Motion, Interaction and Games*, page 41. ACM, 2019.

[148] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014.